ABSTRACT DIGITAL COMPUTERS AND
DEGREES OF UNSOLVABILITY

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

by

Joseph R. Horgan

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Information and Computer Science

Georgia Institute of Technology

August, 1972

ABSTRACT DIGITAL COMPUTERS AND
DEGREES OF UNSOLVABILITY

Approved:

Lucio Chiaraviglio, Chairman

William I. Grosky

Michael D. Kelly

Trevor Evans

Date approved by Chairman: Sept 9 1972

ACKNOWLEDGMENTS

I wish to thank Professor Vladimir Slamecka for his personal encouragement and for assembling a varied and capable faculty. Also, I sincerely thank the members of my guidance and reading committees, Professors John Gwynn, William Grosky, and Michael Kelly, who have helped in the completion of this thesis. Of the other faculty, I would especially like to thank Professor James Gough for his interest in me.

I am grateful to Professor Trevor Evans of Emory University for reading this thesis. Professor Evans's comments on related research influenced the methods of this thesis.

Most especially, I want to thank my advisor, Professor Lucio Chiaraviglio. Professor Chiaraviglio has expertly directed this research, but, more importantly, he has been a model of intellectual vigor and honesty. The best part of my education I owe to Professor Chiaraviglio.

TABLE OF CONTENTS

SUMMARY

The Lindenbaum-Tarski algebra of a formal, infinitary, recursion theory is investigated. This algebra has the upper semi-lattice of degrees of unsolvability as a sub-structure. Quotients of the algebra represent individual degrees, and an algebraic relation among these quotients mirrors the ordering of degrees.

Abstract digital computers are constructed for degrees such that algebraic relations among the computers exactly represent the ordering and join of degrees. In this sense, the theory of degrees of unsolvability is reconstructed in the algebraic theory of abstract digital computers.

CHAPTER I

INTRODUCTION

The breadth of a theory is determined, partially at least, by the theories it comprehends. Among mathematical theories, the theory of Turing machines subsumes the theory of linear bounded automata and ring theory is a subtheory of group theory, which, in turn, is a subtheory of semi-group theory. Among physical theories the theory of quantum mechanics has as a subtheory a theory of light and the theory of continental drift contains a theory of mountain formation. Nor are the social sciences without examples of theories and subtheories; stimulus-response sampling learning theory is supposed to be more general than a certain hierarchical learning theory.

Given a new theory, mathematical or empirical, a first task is to establish its relationship with existing, related theories. We conjecture that the theory of abstract digital computers comprehends a wide variety of computational phenomena. Partial confirmation of that conjecture has been obtained by investigating the relations between the theory of abstract digital computers and the theories of recognizers [29], of Turing machines [10], and of the semantics of programming languages [4]. We investigate in this thesis the relationship of the elementary theory of degrees of unsolvability to the theory of abstract digital computers. The goal of the thesis is to show that the concepts of the theory of abstract digital computers are sufficient to allow the reformulation of the elementary theory of degrees of unsolvability. Such a reformulation is adequate only if the principal relations and operations of the original theory are reflected in the reformulated theory.

The importance of the theories of recognizers, Turing machines and the semantics of programming languages to the information and computer sciences is clear. The immediate importance of the theory of degrees of unsolvability may not be so clear. The elementary theory of degrees of unsolvability explicates the intuitive concept of "relative difficulty of computation" among functions. Two questions of difficulty of computation arise in computer and information science. One is the practical question of the amount of computer space and time necessary to compute a given function. The other is the theoretical question of whether, given the values of some function, another function is then in principle computable. This latter theoretical question is addressed by the theory of degrees. As might be expected, the theoretical question arises about theoretical computing devices. Turing machines with oracles and various formal systems of calculation may be classified into different degrees of unsolvability. Therefore, the elementary theory of degrees treats a significant phenomenon of theoretical computing devices. For this reason the theory of abstract digital computers encompasses a significant dimension of the information sciences if it encompasses the elementary theory of degrees of unsolvability. We demonstrate that the theory of abstract digital computers does indeed encompass the elementary theory of degrees of unsolvability.

The methodology of this thesis supports the assertion of ([23], p. 1) that the study of interesting linguistic systems is possible using the algebraic apparatus of the theory of abstract digital computers. Our formulation of the theory of degrees depends upon a linguistic theory of recursion, R. By investigating the algebra that ensues from R we reduce the linguistic relations among linguistic elements representing functions to algebraic relations among the degrees of the functions. We view the methods of this thesis as paradigmatic of the use of algebraic methods in the study of linguistic systems, and we view the results of the thesis as additional proof that the algebraic theory of abstract digital computers is the appropriate algebraic theory for the algebraic study of computational questions.

Historical Sketch

The history of the theory of abstract digital computers *per se* is short. Even the origins of the theory are recent. Poore [23] is the basic source of the algebraic theory, and the origins of the theory are traced there. It is sufficient here to point out that the algebraic theory is patterned after the theory of Boolean algebras with operators of Jonsson and Tarski [11,12] and, to a certain extent, after the theory of polyadic Boolean algebras of Halmos [8]. The applications of the theory have already been noted. Horgan, Roehrkasse and Chiaraviglio [10] show Turing machines to be among the devices recoverable in the theory of abstract digital computers. Roehrkasse [29] furthermore shows that the classical hierarchy of recognizers, including the salient relations among recognizers, may be reconstructed in the theory of abstract digital computers. It is shown in [10] that abstract digital computers also go beyond automata and Turing machines; they decide the halting problem for Turing machines. DeMillo [4] shows that abstract digital computers can be construed as models, in the logical sense, of programming languages.

The theory of degrees of unsolvability has a somewhat longer history. In 1931 Godel [7] demonstrated that any mathematical system containing arithmetic has formally undecidable questions; that is, questions for which the answers cannot be calculated. Two main issues concerning decidability are: What is it to decide a question? and What is the nature of undecidable questions? The first issue was resolved in 1936 when Post [24,25], Turing [42], Kleene [14], and Church [2] independently proposed formal explications of the informal idea of calculability. These formal concepts have since been demonstrated equivalent [15,17]. Moreover, Church's thesis maintains that whatever one might reasonably mean by calculability is captured by the formal concept. Post's, Turing's, Kleene's, and Church's equivalent formal concepts and Church's thesis are the basis of the investigation of undecidable questions.

Recursive unsolvability, or undecidability, can be demonstrated by two methods. One, a diagonal method, is to construct an undecidable element and follow the attendant <u>reductio</u> <u>ad</u> <u>absurdum</u> argument against decidability. The other method, called reduction, derives the solution of a problem known undecidable from the supposed solution of an open problem. This latter method also proceeds by a <u>reductio</u> argument. Godel's 1931 paper [7] employed the diagonal method to show arithmetic incomplete. Results by Turing [42] and Church [3] give respectively the unsolvability of the halting problem for Turing machines and the undecidability of Peano arithmetic. Turing's result, like Godel's, is constructive, while Church's is reductive. Not until 1947, when Post [27] employed reductive techniques to demonstrate the recursive unsolvability of the so-called problem of Thue, was an outstanding mathematical problem shown unsolvable.

In a familiar setting, the problem of Thue is the word problem for semigroups. Post [27] reduces the halting problem to this word problem, thus proving Thue's problem unsolvable. Post [24,25] first developed reductive techniques in reducing various logics to Post canonical systems. 1944, however, marks Post's most significant methodological contribution [26]: that recursively enumerable sets of positive integers possess a wide variety of decision problems. Primarily from Post's work a number of unsolvability results in number theory and algebra have evolved. Typical is Tarski [39] which gives a decision method for whether or not a polynomial with integral coefficients has simultaneous real roots. Outstanding among the unsolvability results in algebra is the solution by Novikov [22] of the word problem for groups. A finitely presented relation on a group is proved to be undecidable by a constructive argument. Boone's work [1] is an accessible version of this result. Undecidability results are known in a wide variety of algebraic theories, ranging from Mostowski's and Tarski's [19] demonstration of undecidability in the arithmetic of integers and in the theory of rings, to the decidability of certain fields, to Grzegorczyk's [8] chronicle of undecidability for distributive lattices, Brouwerian algebras and related algebraic and geometric systems. The monograph of Tarski, Mostowski and Robinson [40] summarizes many of the results and techniques, mainly reductive, for algebraic theories.

For general formal systems, in the sense of Smullyan [38], somewhat less work exists. Arithmetic is of the lowest undecidable degree, and Fefferman [5] proves that the lowest and each higher tt-degree (that class of systems co-reducible in the sense that every Boolean function satisfied in one is satisfied in the other) have first order and axiomatizable theories. Schoenfield [31] shows each nonrecursive, enumerable degree to contain an axiomatizable, undecidable theory. Much of this kind of work, extended from Post [24,25] and Godel [7], is summarized in Mostowski [16].

Contemporary with Post's early work is the generalization of the Godel-Herbrand concept of recursion by Kleene [14,16]. Kleene [14] introduces the definition of functions by equational systems, a concept central to the research methodology of this paper. An equational system is a formal system for calculating functions by a form of deduction. The notion of one function recursive in, or calculable in, another depends upon Turing's concept of an oracle [42]. A function f is recursive in g if there exists an equational system from which, given the values of g as the oracle, f can be calculated. Degrees are the equivalence classes of functions under the relation "recursive in each other." Because the formal concepts, Turing machine, recursive function and canonical system are equivalent, this notion of degree is perfectly general. Clearly, "recursive in" is a partial ordering among degrees. The first significant structural feature of degrees was discovered by Post and Kleene [18], who found that degrees form an upper semi-lattice. Further, Freidberg [6] and Muchnik [21] prove the existence of incomparable recursively enumerable degrees, solving the problem posed by Post in 1944. Enumerable degrees are dense (Sacks, [33]) and the partial ordering among degrees is highly disconnected: aleph-one pairwise incomparable degrees exist (Schoenfield, [31]). Most important results on degrees appear in Roger's treatise [30] and Sacks' monograph [32].

Plan of Presentation

The strategy of this thesis is as follows. Chapters II and III are parallel in development. Chapter II develops the linguistic systems on which the algebras of Chapter III are based.

Chapter II is devoted entirely to linguistic matters. First, the formation and transformation rules of L(R) are put forth. L(R) is an infinitary propositional language which permits the formation of infinite formulas representing functions in extension. L(R) is shown to be incomplete and consistent. Second, the recursion theory, R, based upon the language L(R) is formed. R is obtained from L(R) by the addition of theorems characterizing the relation "relative recursive" among the formulas representing functions. Also, R is shown to represent accurately the join operation among functions. Like L(R), R is incomplete and consistent.

Chapter III is devoted entirely to algebraic matters. First, the Lindenbaum-Tarski algebra of L(R), called LT(L(R)), is investigated and shown to be a free algebra. Second, the Lindenbaum-Tarski algebra of R, called LT(R), is obtained and demonstrated to be a quotient of LT(L(R)). At this point, a subalgebra A(R) of LT(R) is introduced. A(R) retains all the features of LT(R) appropriate to the study of degrees. We show that A(R) is decomposable as an algebra generated by the union of the Boolean completion, CD, of the semilattice of degrees with the Boolean product of denumerably many isomorphic copies of CD. This decomposition allows us to define in A(R) distinct transition functions for the various degrees and, thus, distinct abstract digital computers for distinct degrees. Finally, the abstract digital computers constructed for the various degrees are shown to have the structure of the degrees of unsolvability.

CHAPTER II

PROPOSITIONAL RECURSION THEORY

Notation

An ordinal is the set of all smaller ordinals, and cardinals are initial ordinals. I is the first transfinite cardinal and I_1 is the second. The cardinal of a set J is indicated $\bar{J}$. At the times we use m and n for cardinals the context will be made clear.

When we speak of functions we ordinarily mean total numeric functions from I^n into I. Functions are construed as ordered pairs. I^n is the Cartesian product of I with itself n-times. Functions of I^n into I are n-ary functions. For simplicity our examples will usually be of 1-ary functions. For sets A and B, A^B is the set of functions of B into A. There should be no confusion over the apparently conflicting definitions of I^n.

The notation for conjunction is '$\cdot$', for disjunction '+', for implication '$\rightarrow$', for biconditional '$\leftrightarrow$'. For infinite conjunction '$\dot{}_{xxx}$' is used where xxx indicates the range of the conjunction, and for infinite disjunction '$+_{xxx}$'. In Boolean algebras the notation for conjunction is used for infimum, disjunction for supremum, infinite conjunction for infinite infimum, and infinite disjunction for infinite supremum. As usual, '$\bigcup_{xxx}$' is set theoretic union over the indexed sets indicated by xxx.

A possible confusion concerns underlined terms. If we underline a term it becomes a numeral, if possible. If not possible, the term is unchanged. A numeral underlined remains a numeral. Some confusion might arise with regard to '$\underline{\leq}$' which always denotes a partial ordering. The same symbol '$\underline{\leq}$' will be used when actually it is an ordering induced from $\leq$. No serious difficulty should occur.

Boolean Algebras

Sikorski [37] is a basic source on Boolean algebras. We review some definitions and theorems for reference or because our notation differs from [37].

2 designates the simple Boolean algebra. Throughout '+' is the finite supremum, '·' the finite infimum, '-' the complement, 0 the least element, 1 the greatest element.

B is complete if for all $D \subseteq B$, $\overset{+}{\underset{A \epsilon D}{}} A$ and $\overset{\cdot}{\underset{A \epsilon D}{}} A$ exist in B. For an infinite cardinal m, we say B is m-complete if for $D \subseteq B$, $\bar{\bar{D}} = m$, then $\overset{+}{\underset{A \epsilon D}{}} A$ and $\overset{\cdot}{\underset{A \epsilon D}{}} A$ exist in B. Thus, as usual, a complete algebra is complete for every m.

B is a regular subalgebra of B' if B is a subalgebra of B' and the suprema and infima of elements in B are the same in B'. If $D \subseteq B$, then $[D]$ is the smallest subalgebra of B containing D. If $[D] = B$ we say that D is a set of generators of B. Then every element of B has the form

$$\overset{+}{\underset{i \leq n}{}} \overset{\cdot}{\underset{j \leq m}{}} d(i,j) D_{i,j}$$

where $D_{i,j} \epsilon D$, $d(i,j)$ is complement indicated (-) or nothing indicated (+), and m depends on n. When G is a class of subsets of B, then B is a

G-algebra if the infimum of every set in G exists in B. If G is the class of all subsets of B, G completeness is completeness. Naturally, a subset D of the G-algebra B is a set of G-generators if the smallest G-subalgebra of B containing D is B itself. The definitions of m-generators and complete generators are analogous.

We say a complete (m-complete) algebra B is the completion (m-completion) of a regular subalgebra B' iff B' is a set of complete generators for B. It is known that completions always exist ([37], p. 153).

We state a theorem for later use.

Theorem 2.1. (Sikorski [37], p. 37). Let h be a one-one mapping of D generating A onto D' generating A', h can be extended to a isomorphism of A onto A' if and only if, for $D_i \varepsilon D$ and $d(i) = (+)$ or $(-)$ for $i \leq n$

$$\dot{\bigcap_{i \leq n}} d(i) D_i = 0_A \text{ iff } \dot{\bigcap_{i \leq n}} d(i) h(D_i) = 0_{A'}.$$

For $A \varepsilon B$ $(A]$, $([A))$, is the principal ideal (principal filter) determined by A. A set of subalgebras $\{C_i\}_{i \varepsilon I}$ of C is independent if and only if for every finite $J \subseteq I$

$$\dot{\bigcap_{j \varepsilon I}} A_j \neq 0$$

for $0 \neq A_j \varepsilon C_j$. Given an independent set of subalgebras $\{C_i\}_{i \varepsilon I}$ of C, $\| \bigcup_{i \varepsilon I} C_i \|$ is the Boolean product of $\{C_i\}_{i \varepsilon I}$, designated $B_{i \varepsilon I} C_i$. Notice that $\| \bigcup_{i \varepsilon I} C_i \|$ is not necessarily an m-algebra if C is. The Boolean product of algebras should not be confused with the direct or Cartesian product of algebras. All further notations and definitions are those of [37].

Recursive Functions and Degrees

A formal characterization of recursive functions requires a formal deductive theory (Kleene, [18]). This theory has the facility for expressing functions as sequences of equations, and the rules of inference of the theory are such that the sequence of equations representing a recursive function is derivable in the theory. The relation of "recursive in" among functions is expressed in the theory by the relation of derivability among the representing sequences of equations of functions.

We describe a modification of Kleene's [18] system which we use as a point of departure for our own formal theory. Call the theory R'. Let it have v_i, $i \epsilon I$ variables, numerals $\underline{i}$, $i \epsilon I$, function symbols $\underline{f}_i$, $i \epsilon I$, and the symbol =. <u>Terms</u> are variables and numerals and function symbols followed by terms. <u>Equations</u> are of the form "$t_1 = t_2$" for terms t_1, t_2. Sequences of equations are countenanced as formulas. The rules of inference are:

R1. Terms may be substituted for variables in equations.

R2. In a sequence of variable free equations, containing both the equation e and $f(\underline{n}_{j_1}, \ldots, \underline{n}_{j_m}) = \underline{n}_k$, we may replace $f(\underline{n}_{j_1}, \ldots, \underline{n}_{j_m})$ by $\underline{n}_k$ in e.

An equation e is <u>derivable</u> from a sequence of equations $e_0, e_1, \ldots, e_n$ if there are equations $e_{n+1}, \ldots, e_{n+m}$ such that e_{n+i}, for $1 \leq i \leq m$, follows from $e_0, e_1, \ldots, e_n, \ldots, e_{n+i-1}$ by R1 and R2 and $e = e_{n+m}$.

A function $f \epsilon I^I$ is <u>general recursive</u> if there exists a sequence of equations $e_1, \ldots, e_n$ such that $f(n) = m$ iff $\underline{f}(\underline{n}) = \underline{m}$ is derivable from $e_1, \ldots, e_n$. A function f is <u>recursive in</u> $f_1, \ldots, f_k$ iff there exists a sequence of equations $e_1,\ldots,e_n$ such that if $f(n) = m$ then $\underline{f}(\underline{n}) = \underline{m}$ is derivable from $e_1,\ldots,e_n$ and the sequences of equations representing $f_1,\ldots,f_k$. Two functions are co-recursive if they are recursive in one another. Notice that there are no sentential connectives in R'. It is therefore not based upon ordinary logic. For this reason R' is not suitable to our needs, as will become evident.

f is equivalent to f' if and only if f and f' are co-recursive. We may partition the class of functions by this equivalence to obtain D, the class of degrees of unsolvability. D may be shown to be an upper semilattice [18]. For $a,b \in D$, $a \leq b$ iff there are $f \in a$ and $f' \in b$ such that f is recursive in f'. a and b also have a least upper bound under $\leq$. If $f \in a$ and $f' \in b$ then the function $f''(i) = 2^{f(i)} \times 3^{f'(i)}$ is called the join of f and f', designated join(f,f'). For some $c \in D$, join(f,f')$\in c$ and we define $a \cup b = c$. $(D, \leq)$ and $(D, \cup)$ are then the same when the ordering is defined in terms of the supremum or vice versa.

As with Boolean algebra for $a \in D$, $(a]$ indicates the principal ideal determined by a, which is the set of all degrees smaller than a. $(a]$ is also called an initial segment of D. Dually, $[a)$ is a final segment of D.

There is more to degrees than we have sketched. Only the elementary theory of degrees has been discussed. We do not discuss recursively enumerable degrees or completions of degrees since we only aim to recover the elementary theory within the theory of abstract digital computers. The methods employed here may lend themselves to a treatment of the entire theory of degrees but that is beyond our present goal.

Languages

For infinite cardinals m and n, $L_{m,n}$ is the first order predicate calculus with equality which permits conjunctions and disjunctions of fewer than m formulas and quantifications involving fewer than n variables (see [13]). $L_{I,I}$ is the ordinary first order predicate calculus with equality, and $L_{I_1,0}$ is like $L_{I,I}$ except that conjunctions and disjunctions of fewer than I_1 formulas are permitted and no quantification is permitted. We shall investigate a fragment L(R) of $L_{I_1,0}$ that permits only certain infinite sets of formulas to have conjunctions.

By a <u>propositional language</u> we mean any language which contains the rules of inference of the ordinary propositional calculus. The languages we deal with are propositional since $L_{I_1,0}$ is propositional.

A language L is an <u>extension</u> of a language L' if and only if every formula of L' is a formula of L. We sometimes say L' is a <u>sub-language</u> of its extension L. If L is an extension of L' and L and L' have the same number of terms and a formula A of L' is a theorem of L iff A is a theorem of L', then L is an <u>inessential extension</u> of L'.

A language T is a <u>theory</u> based upon a language L if T is an extension of L but not an inessential extension. One may form a theory from a language by adding axioms or additional rules of inference.

Given a language L, we can partition the formulas of L, F_L, by the relation A is <u>equivalent to</u> B iff $A \leftrightarrow B$ is a theorem of L. The class of equivalence partitions is in a natural way the carrier of an algebra called the Lindenbaum-Tarski algebra of L and designated LT(L). The class of equivalence partitions of formulas is indicated by $F_L/\leftrightarrow$, sometimes by $L/\leftrightarrow$, and the equivalence class of the formula A is

$|A|_{LT(L)}$, or if not confusing, simply $|A|$. If L is propositional then LT(L) is a Boolean algebra (see Theorem 3.8).

The Lindenbaum-Tarski algebras of certain languages are free algebras. $LT(L_{I,0})$ is free in the class of all Boolean algebras, and $LT(L_{I,I})$ is free in the class of complete Boolean algebras [28]. $LT(L_{I,I})$ is considerably different from $LT(L_{I,0})$ in that quantifiers turn out to correspond algebraically to infinite infima and suprema, i.e.,

$$|(x)(A(x))| = \dot{\bigwedge}_{t \in T} |A(t)|$$

where T is the set of terms of $L_{I,I}$. Of course, in $LT(L_{I,I})$ not every infinite set of elements has an infimum. Only those sets of elements $\{|A(t)| \,|\, A(x)$ is a formula with x free and t is a term$\}$ have infinite infirma. Thus $LT(L_{I,I})$ is a Boolean algebra with partial infinite infimum.

We are concerned with an analagous situation. L(R) is based upon $L_{I_1,0}$. But not all sets of formulas in L(R) have infinite conjunctions. Only those sets which correspond to equations representing a given function in extension have infinite conjunctions. If $\dot{\bigwedge}_{i \in I} A_i$ is an infinite conjunction in L(R), then

$$|\dot{\bigwedge}_{i \in I} A_i| = \dot{\bigwedge}_{i \in I} |A_i|$$

where on the right side '$\dot{\bigwedge}_{i \in I}$' is infinite infimum of the $|A_i|$. Thus in the algebra, only those infinite classes of elements representing constituents of an infinite conjunction have infima.

In the case of $LT(L_{I_1,0})$ we are dealing with a Boolean I-algebra. All conjunctions over I formulas exist in $L_{I_1,0}$; therefore, all infima over I elements exist in $LT(L_{I_1,0})$. However, the primary language of our concern, $L(R)$, has only selected infinite connunctions and, therefore, $LT(L(R))$ is not I-complete. For the most part all the algebras discussed in this thesis are "as complete" as $LT(L(R))$. By this we mean that for morphisms of $LT(L(R))$, subalgebras of $LT(L(R))$ and quotients of $LT(L(R))$ always have infima in the range corresponding to infima in the domain. $LT(L(R))$ is partially complete in much the same way $LT(L_{I,I})$ is (see [37], p. 196).

The following theorem elucidates the relation between sublanguages based on $L_{I_1,0}$ and Lindenbaum-Tarski algebras.

<u>Theorem 2.2</u>. Let L and L' be based on $L_{I_1,0}$. If L' is an inessential extension of L, then $LT(L)$ is a subalgebra of $LT(L')$.

<u>Proof</u>. L' is an inessential extension of L. Hence if $A,B \in F_L$ then, $A \leftrightarrow B$ is a theorem of L iff $A \leftrightarrow B$ is a theorem of L'. Thus a mapping h from $L/\leftrightarrow$ into $L'/\leftrightarrow$ defined

$$h(|A|_L) = |A|_{L'}$$

is well defined and one-one.

h is an embedding.

$$h(-|A|_L) = h(|-A|_L) = |-A|_{L'} = -|A|_{L'} = -(h(|A|_L)) .$$

$$h(|A\cdot B|_L) = |A\cdot B|_{L'} = |A|_{L'}\cdot|B|_{L'} = h(|A|_L)\cdot h(|B|_L) .$$

and

$$h(\dot{}_{i\in I}|A_i|_L) = h(|\dot{}_{i\in I}A_i|_L) = |\dot{}_{i\in I}A_i|_{L'} = \dot{}_{i\in I}|A_i|_{L'} = \dot{}_{i\in I}h(|A_i|_L).$$

Q.E.D.

Abstract Digital Computers

The theory of abstract digital computers is elaborated in Poore [23], and finds application to Turing machines in Horgan, Roehrkasse, and Chiaraviglio [10] and to recognizers in Roehrkasse [29]. We review here only the generalities of the theory but pay extra attention to particularly useful concepts.

An abstract digital computer is an algebra $\underline{B} = (B,A,C)$ where B, the set of "states" of the computer, is a Boolean algebra (hence, the adjective "digital"); $A = B^B$, the set of "actions" of the computer; and $C \epsilon A^B$, the "control unit" of the computer. A process of an abstract digital computer is determined by the iterative action of the control unit on a given state. The sequence

$$C(s_0)(s_0),\ C(s_1)(s_1),\ldots,C(s_i)(s_i),\ldots$$

where $C(s_i)(s_i) = s_{i+1}$ for $i \epsilon I$, is the process of $\underline{B}$ determined by s_0. Because processes of $\underline{B}$ define a transition function on the states of $\underline{B}$

$$T(s_i) = C(s_i)(s_i)$$

we may identify $\underline{B}$ with $(B,T_{\underline{B}})$. Each process of $\underline{B}$ is I long and either periodic, periodic after some non-periodic initial segment, or entirely nonrepeating. We say that $\underline{B}$ halts in n steps on $s_i \epsilon B$ if the process determined by s_i becomes monotonous of period one after n; that is,

$$T_{\underline{B}}^{n-1}(s_i) \neq T_{\underline{B}}^{n}(s_i) = T_{\underline{B}}^{n+1}(s_i) \ .$$

An abstract digital computer is a sort of universal algebra, specifically a Boolean algebra with a single operator; therefore, we are provided with the standard variety of universal algebraic concepts: subalgebra, homomorphism, ideal, etc. These concepts are derived from the corresponding Boolean concepts so as to satisfy some requirements concerning the transition function. In our case the concepts of abstract digital computer ideal and abstract digital computer quotient are of importance. We have some latitude in making these definitions, but we follow Poore ([23], p. 55) essentially.

Given an abstract digital computer $\underline{B} = (B, T_{\underline{B}})$ we say that $M \subseteq B$ is an <u>abstract digital computer ideal</u> of $\underline{B}$ iff M is an ideal of B and for $A, C \in B$ if

$$((-A)\cdot C) + (A\cdot(-C)) \in M \quad \text{then } T_{\underline{B}}(A) = T_{\underline{B}}(C).$$

The definition merely stipulates that if $|C| = |A| \in B/M$ then

$$T_{B/M}(|A|) = T_{B/M}(|C|)$$

since

$$T_{B/M}(|A|) = |T_{\underline{B}}(A)| = |T_{\underline{B}}(C)| = T_{B/M}(|C|) \ .$$

Therefore a <u>quotient abstract digital computer</u> is an abstract digital computer $\underline{B}/M = (B/M, T_{B/M})$ where M is an abstract digital computer ideal of $\underline{B}$.

Obviously, h is an <u>abstract digital computer homomorphism</u> of the abstract digital computer (A, T_A) into the abstract digital computer (B, T_B) if h is a Boolean homomorphism of A into B and for $a \in A$

$$h(T_A(a)) = T_B(h(a)) \ .$$

Also, if a transition function T_A on a Boolean algebra A is a partial function, then we call (A,T_A) a partial abstract digital computer. Clearly, such a partial transformation function T_A may be extended to a total function by defining it to be identity where undefined. We will find partial abstract digital computers of use in the sequel.

The information we need about the theory of abstract digital computers is contained in the definitions given above, so we refer the reader to the previously cited references for further information.

The Language L(R)

By the language L(R) we mean one based on the infinitary predicate calculus, $L_{I_1,0}$. The most outstanding feature of L(R) is that it permits some infinite conjunctions and disjunctions of length I. For the purpose of distinguishing infinite conjunctions, for example, from finite conjunctions we use the notation '$\dot{}_{i\varepsilon I}$' where the index set I is countably infinite. Occasionally we may also use the notation '$\dot{}_{i\leq n}$' or '$\dot{}_{i\varepsilon J}$' to indicate the finite conjunction of n or $\bar{J}$ formulas, but in those cases J will be finite and $n \leq I$. These latter uses of indexed conjunction may be viewed as abbreviations for the sequential conjunctions of the constituent formulas, while the infinitely indexed conjunction is not an abbreviation and must be introduced as a separate connective.

Formation Rules of L(R).

i) The logical connectives of L(R) are:

- (negation), $\cdot$ (conjunction), + (disjunction), $\dot{}_{i\varepsilon I}$ (infinite conjunction over I), $\overset{+}{}_{i\varepsilon I}$ (infinite disjunction over I).

We use the standard abbreviations representing implication and biconditional, and we observe the infinitary DeMorgan laws:

$$-(\dot{\underset{i\epsilon I}{}} A_i) = \overset{+}{\underset{i\epsilon I}{}} (-A_i)$$

$$-(\overset{+}{\underset{i\epsilon I}{}} A_i) = \dot{\underset{i\epsilon I}{}} (-A_i)$$

The interpretation of these connectives is standard for the finitary connectives and the infinite connectives may be interpreted in the obvious infinite extension of truth tables. These matters are discussed in Karp [13].

L(R) of course must have the usual <u>punctuation</u>

ii) ((left parenthesis), and) (right parenthesis).

Punctuation will be used as the context dictates, although strict adherence to the formation rules of L(R) might require otherwise.

There is also one special predicate, <u>equality</u>.

iii) = (equality)

We understand its meaning to be governed by the usual axioms for equality; reflexivity, symmetry, transitivity, and the substitutivity of equality, which plays an important role in the deductive system R to be considered later.

We have the usual number of <u>variables</u>, one for every element of I.

iv) v_i, $i \epsilon I$

These are individual variables since L(R) is a first order infinitary language. Also, because the theory for which L(R) is the language is a kind of arithmetic, L(R) has the usual complement of numerals, one for every number in I. $\underline{I}$ will designate this set of <u>numerals</u>, ordered in the natural way, and $\underline{i}$ will indicate the i^{th} numeral, which names the number i.

v) $\underline{I} = (\underline{i} \mid i \in I)$

We shall indicate numerals by underlining numerals, generally.

The last categories of the alphabet of L(R) are two disjoint sets of function symbols. One set is the set of <u>auxiliary functors</u>, I of them of each finite adicity

vi) $(h_i^n \mid i \in I)$, $n \in I$

Here $i \in I$ indexes the i^{th} functor of adicity n.

We treat the second set of functors, which we call <u>numeric functors</u>, separately because only selected formulas involving these functors will have infinite conjunctions and disjunctions under the rules of formation. In fact, these numeric functors are constants designating functions of I^n into I. Let

$$(1) \qquad (f_i^n \mid i \in I_1)$$

for a particular $n \in I$ be a one-one enumeration of the functions of I^n into I. Then

vii) $(\underline{f}_i^n \mid i \in I_1)$

is a sequence of functional constants such that $\underline{f}_i^n$ designates the i^{th} n-adic function in the enumeration (1). Thus we have functional constants which designate uniquely all the total functions of natural numbers. The enumerations (1) of the total numeric functions require the axiom of choice.

Our purpose in providing L(R) with constants for functions is to eventually fix the interpretation of computations concerning functions. The special role of these numeric constants will become evident in the formation rules of L(R), to which we turn.

The formation rules of L(R) are chosen, as has been mentioned, to allow the construction of a propositional, deductive theory of recursive functions.

The class of <u>terms</u> of L(R) is the least class closed under the following rules:

i) numerals and variables are terms;

ii) If $\underline{f}_i^n$ for $i \in I_1$ is an n-adic functional constant and $t_1, \ldots, t_n$ are terms, then $\underline{f}_i^n(t_1, \ldots, t_n)$ is a term.

iii) If h_i^n for $i \in I$ is an n-adic auxiliary functor and $t_1, \ldots, t_n$ are terms, then $h_i^n(t_1, \ldots, t_n)$ is a term.

We define <u>numeric terms</u> as terms of L(R) which involve only numerals and functional constants.

The class of <u>well-formed formulas</u>, wff's, is the least class closed under the following rules:

i) If t_1 and t_2 are terms, then the <u>equation</u> $t_1 = t_2$ is a wff;

ii) If A and B are wff's, then so are $-A$, $A + B$, and $A \cdot B$;

III) If $t_1, \ldots, t_n$ are numeric terms, then ${}_{i \in I}\dot{}A_i$ is a wff, where for each $i \in I$ A_i is the <u>numeric equation</u> $\underline{\underline{f}_i^n(\underline{t}_1, \ldots, \underline{t}_n) = f_i^n(t_1, \ldots, t_n)}$ for some particular $i \in I_1$.

Several points should be noted about iii). First, if $t_1,\ldots,t_n$ are numeric terms, then $\underline{t}_1,\ldots,\underline{t}_n$ are numerals since each of $t_1,\ldots,t_n$ designates a number. Furthermore, $\underline{f}_i^n(\underline{t}_1,\ldots,\underline{t}_n) = \underline{f_i^n(t_1,\ldots,t_n)}$ makes sense, because the left-hand side is a numeric term expressing the value of the function named by $\underline{f}_i^n$ at the numbers named by $\underline{t}_1,\ldots,\underline{t}_n$ which is the number $f_i^n(t_1,\ldots,t_n)$. Also notice that although we permit infinite conjunctions of appropriate formulas, infinite disjunctions will exist only as the DeMorgan transformations of the negations of these conjunctions. Last, by iii) we are allowing only infinite conjunctions over numeric equations expressing some one numeric function in extension. This is possible because of the introduction of functional constants for each numeric function. Our language only needs infinite expressions representing functions in extension; therefore, we restrict formation of infinite expressions to just those expressions. This restriction also facilitates our algebraic considerations at a later point.

Properly $L(R)$ should be called an applied language because it has functional constants with fixed interpretations. The formation of $L(R)$ is similar to the formation of the first-order predicate calculus with equality obtained by adding the equality predicate and governing axioms to the predicate calculus. The difference with $L(R)$ is that it is formed from $L_{I_1,0}$ with equality by adding the functional constants.

Clearly, I^n can be well-ordered isomorphically to I. For instance, the natural ordering $(n_1,\ldots,n_m) < (k_1,\ldots,k_m)$ if and only if for the least $1 \leq i \leq m$ such that $n_i \neq k_i$, $n_i < k_i$, is one. Henceforth, for convenience we shall write $f_i^n(j)$, $\underline{f}_i^n(\underline{j})$, etc., where j is the j^{th} member of the ordering of n-typles of I or $\underline{I}$, etc., and $\underline{j}$ designates the n-tuple of numerals corresponding to the j^{th} n-tuple of numbers. Hereafter, we shall always consider the conjunction $\dot{}_{i\varepsilon I}A_i$ ordered by the expedient that $A_i = (\underline{f}^n(\underline{i}) = \underline{f^n(i)})$. That is, the i^{th} constituent of $\dot{}_{i\varepsilon I}A_i$ is the equation involving the i^{th} n-tuple in the ordering of I^n. We will drop the superscript of f^n where such omissions detract nothing from the meaning.

Transformation Rules of L(R)

The rules of transformation given here for L(R) are essentially similar to the appropriate subset of those given for $L_{I_1,I}$ by Scott [36]. We differ from him in directly defining the class of theorems of L(R) rather than positing axioms and rules of inference. From our eventual algebraic point of view there is no difference; indeed, from that point of view the theorems can be obtained in any way whatsoever.

In the following definition '$A \rightarrow B$' means as usual '$(-A) + B$'. The class of theorems of L(R) is the least class closed under the following rules; for formulas A, B, C and $\dot{}_{i\varepsilon I}A_i$.

i) $A \rightarrow (B \rightarrow A)$, $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$, and $((-B) \rightarrow (-A)) \rightarrow (A \rightarrow B)$ are theorems;

ii) If $j \varepsilon I$, then $(\dot{}_{i\varepsilon I}A_i) \rightarrow A_j$ is a theorem;

iii) If $A \rightarrow B$ and A are theorems, then so is B;

iv) If $B \rightarrow A_i$ for $i\varepsilon I$ are theorems, then $B \rightarrow (\dot{}_{i\varepsilon I}A_i)$ is a theorem;

v) If t_1 is a term, then $t_1 = t_1$ is a theorem;

vi) If $A(t_1)$ is a wff differing from $A(t_2)$ only in having the term t_1 whenever $A(t_2)$ has the term t_2, then

$(A(t_2) \cdot t_1 = t_2) \rightarrow A(t_1)$ is a theorem;

vii) If $A(\underline{i})$ is a wff differing from $A(v_1)$ only in having the numeral $\underline{i}$ whenever $A(v_1)$ has the variable v_1, then $A(v_1) \rightarrow A(\underline{i})$ is a theorem.

Rules i) and iii) give the theorems of the propositional calculus, and hence the wff's of L(R) may be quotiented into a Boolean algebra by biconditional equivalence (Tarski and Scott [41]). This feature of L(R) is, of course, fundamental to our purpose of finding the Boolean structure of R. Rule ii) guarantees that infinite conjunctions, when they exist, imply all their constituents, as is the case with ordinary conjunctions. Rule iv) expresses the infinite version of the familiar adjunctive property of conjunctions. Rules v) and vi) are respectively the reflexivity and substitutivity of equality. Rule vi) serves in R as Kleene's rule of substitution does in the ordinary calculus for recursive functions. The other of his rules of inference, replacement of variables by numerals, has its counterpart for us in Rule vii). In L(R), although there is actually no quantification, vi) and vii) together give us approximately the facility that universal specification lends to a first-order language with quantification.

We end our investigation of L(R) by examining some of its metamathematical properties. Only one, consistency, is of immediate interest--an inconsistent language cannot house a consistent theory--but the other, incompleteness, tells us that the algebras we will study are not simple.

<u>Theorem 2.3</u>. L(R) is consistent and incomplete.

<u>Proof</u>. $B = (\underline{f}_1(\underline{i}) = \underline{f_1(i)})$ is true under any well ordering of the

functions of I into I and corresponding assignment of functional constants. Clearly, B is not a theorem of L(R). It is an atomic formula; thus it would necessarily have to follow from rules iii) or v). v) is impossible since the term '$f_1(\underline{i})$' is distinct from the term '$\underline{f_1(i)}$'. Then suppose iii) applies. If so, A $\rightarrow$ B and A are theorems for some A. But then 'A $\rightarrow$ B' must have the form evident in rule ii); that is, A is an infinite conjunction. If A is a theorem it must have followed from rule iv) involving theorems C and C $\rightarrow$ A. But then C $\rightarrow$ B must be a theorem; thus we must apply rule ii) again and we see that by _reductio ad impossibile_ we can never "detach" B. Therefore, L(R) has a true formula which is not a theorem.

Q.E.D.

L(R) is a consistent propositional language. Those two properties are the primary _desiderata_ of the language upon which R is to be based.

The Theory R

The theory R is the infinitary propositional recursion theory based upon the language L(R). The relation which we wish to capture in R is the relation of relative recursiveness between formulas representing functions. Thus, the principal criterion of adequacy for R is that:

A1. If f is recursive in g then the conditional with the formula representing g as antecedent and the formula representing f as consequent is a theorem of R.

A second adequacy criterion for R is that:

A2. In R the conjunction of the formulas representing functions f and g is equivalent to the formula representing the join of f and g.

One might suppose that those formulas representing recursive functions should be theorems of R. While it is possible to construct R in this way, the eventual result is a rougher algebraic theory. In any case, there is no necessity in this supposition and it will not hold for R. R is constructed from L(R) by a modification to the rules of theoremhood of L(R).

Before stating the modified definition of theorem we need some preliminary definitions. A formula of L(R) is <u>essentially finite</u> if and only if it is not equivalent to a formula that has a formula given by formation rule iii) as a subformula. A <u>subformula</u> is a formula occurring as a consecutive string of alphabetic symbols in another formula. If a formula is not essentially finite, it is said to be <u>essentially infinite</u>. Let A be an essentially infinite formula and B be an essentially finite formula; then A and B are <u>compatible</u> if and only if when $A \cdot B \rightarrow (\underline{f}^n_i(\underline{j}) = \underline{k}) \cdot (\underline{f}^n_i(\underline{j}) = \underline{m})$ is a theorem then $k = m$. Now we can give the definition of theorem in R.

The class of <u>theorems of R</u> is the least class of formulas of L(R) closed under the rules i) through vii) for theorems in L(R) and the additional rules:

viii) If B is a finite conjunction of equations compatible with $\dot{}_{i\epsilon I}A_i$ and $((\dot{}_{i\epsilon I}A_i) \cdot B) \rightarrow C_i$ is a theorem for every $i \epsilon I$, then $(\dot{}_{i\epsilon I}A_i) \rightarrow (\dot{}_{i\epsilon I}C_i)$ is a theorem.

ix) If $(\dot{}_{i\epsilon I}A_i) \rightarrow (\dot{}_{i\epsilon I}B_i)$ is a theorem, then $A_i \rightarrow B_i$ is a theorem for $i \epsilon I$.

Rule viii) is an obvious paraphrase of Kleene's definition of relative recursive, where a function f is recursive in a function g if and only if there exists a finite system of equations E (our B) such that given the infinite sequence of equations representing g (our $\dot{_{i\epsilon I}}A_i$), all the equations representing f (our $\dot{_{i\epsilon I}}C_i$) are derivable using substitution of equals for equals (our theorem rule vi)) and replacement of variables by numerals (our theorem rule vii)). Rule ix) is merely a technical convenience which takes effect to make the algebraic theory smoother. We may recall the convention that $A_i = (\underline{f}(\underline{i}) = \underline{f(i)})$ for some constant f. Rule ix) does say that $A_i \rightarrow C_i$ for $i \epsilon I$ are theorems when $\dot{_{i\epsilon I}}A_i \rightarrow \dot{_{i\epsilon I}}C_i$ is a theorem obtained by viii). Moreover, ix), depending on viii) as it does, can be of no harm to the consistency of R if viii) is not. We must show that R meets our two criteria of adequacy; that conjunction may be construed as join, and implication serves as the converse of the relation recursive in. Of course, R must be consistent.

<u>Theorem 2.4</u>. R is consistent and incomplete.

<u>Proof</u>. L(R) is consistent and R differs from L(R) only by the addition of rules viii) and ix), so we need only show that they introduce no inconsistency. Actually the argument of 2.3 may be modified for this purpose. Again consider B as in 2.3. Again because it is atomic it would have to follow from iii) or v), and again it cannot be v). If B is a theorem by iii), for some theorem A, $A \rightarrow B$ must be a theorem. Now '$A \rightarrow B$' must have the form of ii) or ix). In the first instance, A is an infinite conjunction and we have another <u>reductio</u>. In the

second instance, A must be another atomic formula which cannot be detached on the same grounds B could not be.

Q.E.D.

The following theorems establish that R meets the adequacy criteria A1 and A2. Theorem 2.5 establishes A1.

<u>Theorem 2.5</u>. If for all $i \in I$, $A_i = (\underline{f}_j(\underline{i}) = \underline{f_j(i)})$ and $C_i = (\underline{f}_k(\underline{i}) = \underline{f_k(i)})$, then ${}^{\cdot}_{i\in I}A_i \rightarrow {}^{\cdot}_{i\in I}C_i$ is a theorem of R iff the function f_k is recursive in the function f_j.

<u>Proof</u>. Suppose ${}^{\cdot}_{i\in I}A_i \rightarrow {}^{\cdot}_{i\in I}C_i$ is a theorem of R. Then it must be so in virtue of rule viii); thus there exists some essentially finite conjunction of equations B compatible with ${}^{\cdot}_{i\in I}A_i$ such that $({}^{\cdot}_{i\in I}A_i)\cdot B \rightarrow C_i$ is a theorem of L(R) for every $i \in I$. From an examination of the definition of theorem in L(R) only rules ii), vi) and vii) can be essentially involved in the deduction of $({}^{\cdot}_{i\in I}A_i)\cdot B \rightarrow C_i$. If $C_i = A_i$ for every i, then $({}^{\cdot}_{i\in I}A_i)\cdot B \rightarrow C_i$ follows from ii), and we are done since every function is recursive in itself. If $C_i \neq A_i$ then $({}^{\cdot}_{i\in I}A_i)\cdot B \rightarrow C_i$ can follow by simplification for a finite number of $i \in I$ where C_i is a constituent of B, but vi) and vii) must be employed for the remaining infinite number of theorems, $({}^{\cdot}_{i\in I}A_i)\cdot B \rightarrow C_j$ where $j \in I$ and C_j is not a constituent of B. However, vi) and vii) are just the analogs of Kleene's rules of substitution and replacement. Therefore, we see that since B, an essentially finite conjunction of equations, can be identified with a system of equations, the first half of the theorem follows, by identifying ${}^{\cdot}_{i\in I}A_i$ with the infinite system of equations representing f_j, identifying ${}^{\cdot}_{i\in I}C_i$

with the system of equations representing f_k, and B with the auxiliary system of equations.

The converse argument is just as obvious. The systems representing f_j and f_k and the auxiliary system E can be identified with ${}_{i\dot{\in}I}A_i$, ${}_{i\dot{\in}I}C_i$ and B respectively. If replacement and substitution suffice in Kleene's formalism for the derivation of $\underline{f}_k(\underline{i}) = \underline{f_k(i)}$ for all i, then vi), vii) and ii) are sufficient to show $(({}_{i\dot{\in}I}A_i)\cdot B) \rightarrow {}_{i\dot{\in}I}C_i$ to be a theorem in R.

Q.E.D.

Theorem 2.6 establishes A2.

<u>Theorem 2.6</u>. Let $A_i = (\underline{f}_j(\underline{i}) = \underline{f_j(i)})$, $B_i = (\underline{f}_k(\underline{i}) = \underline{f_k(i)})$ and $C_i = (\underline{f}_m(\underline{i}) = \underline{f_m(i)})$ for all $i \in I$, then f_m and join(f_j, f_k) are co-recursive if and only if $(({}_{i\dot{\in}I}A_i)\cdot({}_{i\dot{\in}I}B_i)) \leftrightarrow ({}_{i\dot{\in}I}C_i)$ is a theorem of R.

<u>Proof</u>. If f_m and join(f_j, f_k) are co-recursive, then f_m is recursive in join(f_j, f_k) which is, in turn, recursive in f_m. join(f_j, f_k) we recall is a function f_p such that $f_p(i) = 2^{f_j(i)} \times 3^{f_k(i)}$. Therefore there exist systems of equations E_1 and E_2 such that from $\underline{f}_m(\underline{1}) = \underline{f_m(1)}$, $\underline{f}_m(\underline{2}) = \underline{f_m(2)}, \ldots$ and E_1, $\underline{f}_p(\underline{i}) = \underline{f_p(i)}$ is derivable for all $i \in I$, and from $\underline{f}_p(\underline{1}) = \underline{f_p(1)}$, $\underline{f}_p(\underline{2}) = \underline{f_p(2)}, \ldots$ and E_2, $\underline{f}_m(\underline{i}) = \underline{f_m(i)}$ is derivable for every $i \in I$. By Kleene's definition of recursive in, E_1 and E_2 must be finite sequences of equations whose conjunctions in R are compatible with ${}_{i\dot{\in}I}C_i$ and ${}_{i\dot{\in}I}(\underline{f}_p(\underline{i}) = \underline{f_p(i)})$ in R. Thus

$$(1) \qquad ({}_{i\dot{\in}I}C_i) \leftrightarrow {}_{i\dot{\in}I}(\underline{f}_p(\underline{i}) = \underline{f_p(i)})$$

is a theorem of R.

f_p is clearly recursive in f_j, f_k, defined as it is by primitive recursion from f_j and f_k. Thus there exists some finite system of equations E_3 such that its conjunction in R is compatible with $(\dot{}_{i\varepsilon I}A_i)\cdot(\dot{}_{i\varepsilon I}B_i)$, hence

(2) $$((\dot{}_{i\varepsilon I}A_i)\cdot(\dot{}_{i\varepsilon I}B_i)) \rightarrow \dot{}_{i\varepsilon I}(\underline{f}_p(\underline{i}) = \underline{f_p(i)})$$

is a theorem of R. On the other hand, f_j and f_k are each recursive in f_p. So, there are finite systems of equations E_4 and E_5 used in the computations. However, the conjunction of E_4 and the conjunction of E_5 may have conflicting auxiliary equations. We avoid this difficulty by modifying E_4 to E_4' and E_5 to E_5' such that they share no auxiliary function symbols. Then the conjunction of all of the equations in E_4' and E_5' is compatible with $\dot{}_{i\varepsilon I}(\underline{f}_p(\underline{i}) = \underline{f_p(i)})$ in R. Therefore,

(3) $$\dot{}_{i\varepsilon I}(\underline{f}_p(\underline{i}) = \underline{f_p(i)}) \rightarrow ((\dot{}_{i\varepsilon I}A_i)\cdot(\dot{}_{i\varepsilon I}B_i))$$

is a theorem in R. (1), (2) and (3) establish the sufficiency of our theorem.

The necessity of the theorem must also be established. Let f_p be as before. We know f_p to be recursive in f_j, f_k. Since $(\dot{}_{i\varepsilon I}C_i) \rightarrow ((\dot{}_{i\varepsilon I}A_i)\cdot(\dot{}_{i\varepsilon I}B_i))$ is a theorem there is a finite conjunction D of equations in R compatible with $\dot{}_{i\varepsilon I}C_i$ such that $(\dot{}_{i\varepsilon I}C_i)\cdot D \rightarrow ((\dot{}_{i\varepsilon I}A_i)\cdot(\dot{}_{i\varepsilon I}B_i))$ is a theorem in R. Then D represents a set of equations which when sequenced differently can serve as systems of equations E_1 and E_2 such that from $\underline{f}_m(\underline{1}) = \underline{f_m(1)}$, $\underline{f}_m(\underline{2}) = \underline{f_m(2)}, \ldots$ and E_1, $\underline{f}_j(\underline{i}) = \underline{f_j(i)}$ can be derived for all $i\varepsilon I$, and from $\underline{f}_m(\underline{1}) = \underline{f_m(1)}$, $\underline{f}_m(\underline{2}) = \underline{f_m(2)}, \ldots$ and E_2,

$\underline{f}_k(\underline{i}) = \underline{f_k(i)}$ can be derived for all $i \in I$. In other words, f_k and f_j are each recursive in f_m. But we know f_p is recursive in f_k, f_j. Thus f_p is recursive in f_m.

The other way, if $(\dot{\ }_{i\in I}A_i)\cdot(\dot{\ }_{i\in I}B_i) \rightarrow \dot{\ }_{i\in I}C_i$ is a theorem in R, then for some compatible E_3, $((\dot{\ }_{i\in I}A_i)\cdot(\dot{\ }_{i\in I}B_i)\cdot E_3) \rightarrow \dot{\ }_{i\in I}C_i$ is a theorem in R. Therefore f_m is recursive in f_k, f_j. But again, we know f_k and f_j to be recursive in f_p, thus f_m is recursive in f_p. We have established the co-recursivity of f_m and f_p, the necessity of the theorem, and the theorem.

Q.E.D.

A simple theorem will later prove useful.

<u>Theorem 2.7</u>.

$$(\dot{\ }_{i\in I}A_i)\cdot(\dot{\ }_{i\in I}B_i) \leftrightarrow \dot{\ }_{i\in I}(A_i\cdot B_i)$$

is a theorem of R.

<u>Proof</u>. Obviously, for $i \in I$

$$(\dot{\ }_{i\in I}A_i)\cdot(\dot{\ }_{i\in I}B_i) \rightarrow A_i\cdot B_i$$

thus by rule iv),

$$(\dot{\ }_{i\in I}A_i)\cdot(\dot{\ }_{i\in I}B_i) \rightarrow \dot{\ }_{i\in I}(A_i\cdot B_i) .$$

Conversely, for $i \in I$

$$\dot{\ }_{i\in I}(A_i\cdot B_i) \rightarrow A_i$$

$$\dot{\ }_{i\in I}(A_i\cdot B_i) \rightarrow B_i$$

thus by iv)

$$\dot{\bigwedge_{i\in I}}(A_i \cdot B_i) \rightarrow \dot{\bigwedge_{i\in I}} A_i$$

$$\dot{\bigwedge_{i\in I}}(A_i \cdot B_i) \rightarrow \dot{\bigwedge_{i\in I}} B_i$$

so

$$\dot{\bigwedge_{i\in I}}(A_i \cdot B_i) \rightarrow (\dot{\bigwedge_{i\in I}} A_i)\cdot(\dot{\bigwedge_{i\in I}} B_i)\,.$$

Q.E.D.

These are the last of our essentially linguistic considerations. R is an infinitary, propositional recursion theory and therefore its Lindenbaum-Tarski algebra is a Boolean algebra which reveals the structure of two primary aspects of recursion theory: the nature of relative recursion among functions and the join operation of functions. We now turn to an investigation of LT(L(R)).

CHAPTER III

THE ALGEBRA OF R AND DEGREES

The Algebra LT(L(R))

The algebra LT(L(R)), the Lindenbaum-Tarski algebra of the language L(R), is fundamental to our investigations. We enter the following theorem which is a slight modification of theorem 10.6 in ([28], p. 250) without proof.

Theorem 3.8. (Rasiowa, Sikorski [28]). If a language L is propositional, i.e., if theorem rules i and iii of L(R) are theorem rules of L, then LT(L) is a Boolean algebra and

i) $|A| \leq |B|$ iff $A \to B$ is a theorem of L.

ii) $|A| \cdot |B| = |A \cdot B|$.

iii) $-|A| = |-A|$.

iv) $|A| + |B| = |A + B|$.

v) $|A| \to |B| = |A \to B|$.

Theorem 3.9. LT(L(R)) is a Boolean algebra.

Proof. i) and iii) of L are rules of theoremhood in L(R); then L(R) is propositional and, by 3.8, LT(L(R)) is Boolean. It must be shown that in LT(L(R))

$$\left|\dot{\prod_{i \in I}} A_i\right| = \dot{\prod_{i \in I}} |A_i|$$

where on the left side '$\dot{}_{i\varepsilon I}$' is infinite conjunction, and on the right hand side '$\dot{}_{i\varepsilon I}$' is the infimum over $|A_i|$ in LT(L(R)) for $i\varepsilon I$.

Theorem 3.10. In LT(L(R)), $|\dot{}_{i\varepsilon I}A_i| = \dot{}_{i\varepsilon I}|A_i|$.

Proof. By rule ii) of L, for every $i\varepsilon I$

$$(\dot{}_{i\varepsilon I}A_i) \rightarrow A_i$$

is a theorem. Thus, for every $i\varepsilon I$

$$|\dot{}_{i\varepsilon I}A_i| \leq |A_i|$$

and

$$|\dot{}_{i\varepsilon I}A_i| \leq \dot{}_{i\varepsilon I}|A_i| \ .$$

The other way, since

$$\dot{}_{i\varepsilon I}|A_i| \ \varepsilon\ LT(L(R))$$

there is some B in L(R) such that

$$|B| = \dot{}_{i\varepsilon I}|A_i| \ .$$

But then for $i\varepsilon I$,

$$|B| \leq |A_i| \ .$$

Thus, for $i\varepsilon I$

$$B \rightarrow A_i$$

is a theorem, and by rule iv)

$$B \rightarrow (\dot{}_{i\varepsilon I} A_i)$$

is a theorem, thus

$$|B| \leq |\dot{}_{i\varepsilon I} A_i| ,$$

and the theorem is proved.

Q.E.D.

LT(L(R)) can now be viewed as a Boolean algebra with a partial infinitary operation.

$$L(R)/\leftrightarrow, \cdot, -, 0, 1, \dot{}_{i\varepsilon I})$$

where the partial infinite infimum of LT(L(R)) has a role similar to the partial infinite infimum given by quantification in the Lindenbaum-Tarski algebra of the predicate calculus. Fortunately, LT(L(R)) is free.

It is well known that the Lindenbaum-Tarski algebra of LT($L_{I,0}$) is free in the class of all complete Boolean algebras ([28], p. 261). We use this fact to prove that LT(L(R)) is free in the class of all complete Boolean algebras. First we prove a simple theorem.

<u>Theorem 3.11</u>. LT($L_{I_1,0}$) is free in the class of all complete Boolean algebras, and the set of elements $E = \{|A| \mid A$ is an atomic formula$\}$ is a set of free generators.

<u>Proof</u>. $L_{I,0}$ is a sublanguage of $L_{I_1,0}$; therefore, by 2.3 LT($L_{I,0}$) is a subalgebra of LT($L_{I_1,0}$). The generators of LT($L_{I,0}$) are identical to the generators of LT($L_{I_1,0}$); therefore any mapping of E into a complete Boolean algebra may be extended to a homomorphism of LT($L_{I_1,0}$).

Q.E.D.

Now we can prove the main theorem.

Theorem 3.12. LT(L(R)) is free in the class of complete Boolean algebras, and the set of elements $E = \{|A| \mid A$ is an equation$\}$ is a set of free generators.

Proof. Extend $L_{I_1,0}$ by adding the class of functors of the formation rule vii) of L(R) and the class of numerals of formation rule v). Call this new language $L_{I_1,0}^+$. $LT(L_{I_1,0}^+)$ is clearly free in the class of complete Boolean algebras with a larger set of free generators than $LT(L_{I_1,0})$. $L_{I_1,0}^+$ is an inessential extension of L(R) since it only allows some additional infinite conjunctions and disjunctions; therefore, by 2.1 LT(L(R)) is a subalgebra of $LT(L_{I_1,0}^+)$. Furthermore, it is evidently a regular subalgebra, and the set of generators of $LT(L_{I_1,0}^+)$ contains the set of generators of LT(L(R)). Therefore, any mapping of the generators of $LT(L_{I_1,0}^+)$ into a complete Boolean algebra is extensible to a homomorphism h, and h restricted to LT(L(R)) is a homomorphism.

Q.E.D.

The statement of 3.11 may be modified to exchange m-complete for complete, where m is the cardinal equal to the number of terms in $L_{I_1,0}$. In the case of $L_{I_1,0}$, m = I, but in the case of $L_{I_1,0}^+$, m = I_1. All that the completeness of the range algebra assures is that infima exist corresponding to infinite infima in LT(L(R)). The relation between the free algebra LT(L(R)) and LT(R) is that of algebra to quotient algebra, since LT(L(R)) is free and R is an extension of L(R), as we shall soon see.

The Algebra LT(R)

R is an extension of the language L(R). However, R has theorems formed in L(R) which are not theorems of L(R); thus R is not an inessential extension of L(R). LT(R) also is not a subalgebra of LT(L(R)). Rather it is a quotient of LT(L(R)). As might be expected, the free generators of LT(L(R)), although generators of LT(R), are not free for LT(R).

Theorem 3.13. LT(R) is not free on E in the class of complete algebras.

Proof. Where f_n is recursive in f_m and $A_i = (\underline{f}_m(\underline{i}) = \underline{f_m}(i))$ and $B_i = (\underline{f}_n(\underline{i}) = \underline{f_n}(i))$, then $\dot{\bigwedge}_{i\in I} A_i \rightarrow \dot{\bigwedge}_{i\in I} B_i$ by 2.5 is a theorem. Consequently, so is

$$A_i \rightarrow B_i$$

by rule ix) of R. Thus by 3.8

$$|A_i| \leq |B_i| \;.$$

Consider a mapping r into 2 such that

$$r(|A_i|) = 1 \text{ and } r(|B_i|) = 0 \;.$$

r can not be extended to a homomorphism since

$$r(|A_i|) > r(|B_i|) \;.$$

Q.E.D.

As stated, LT(R) is a quotient of LT(L(R)). Two methods of proof are open to us. The first relies upon the fact that R is a theory based upon the language L(R). The representatives in LT(R) of theorems of R constitute a filter, T, in LT(L(R)) and the quotient of LT(L(R)) by T (or by the dual of T) is isomorphic to LT(R). This method is fully discussed in ([28], p. 273). We choose a more algebraic proof using 3.12, relying on the fact that LT(L(R)) is free.

<u>Theorem 3.14</u>. LT(R) is isomorphic to a quotient of LT(L(R)).

<u>Proof</u>. By 3.12, LT(L(R)) is free in the class of complete Boolean algebras. Consider the completion of LT(R) which we denote CLT(R). Define the mapping h from the generators $E = \{|A|_L \mid A \text{ is an equation of } L(R)\}$ of LT(L(R)) into CLT(R) as follows:

$$h(|A|_L) = |A|_C$$

(We signify elements in LT(L(R)) by $||_L$, in CLT(R), by $||_C$ and in LT(R), by $||_R$). If A is an atomic formula of L(R), then it is an atomic formula of R. Therefore, since LT(R) is a regular subalgebra of CLT(R), implying $|A|_R = |A|_C$, h maps the generators of LT(L(R)) onto the generators of LT(R).

h is a mapping, for suppose that

$$|A|_L = |B|_L$$

then $A \leftrightarrow B$ is a theorem of L(R) and a theorem of the extension, R, of L(R).

Necessarily, then

$$|A|_C = |A|_R = |B|_R = |B|_C$$

so

$$h(|A|_L) = h(|B|_L) .$$

Then h is extensible to a homomorphism h^* of LT(L(R)) into CLT(R). But h^* maps the generators of LT(L(R)) onto the generators of LT(R); therefore,

$$h^*(LT(L(R))) = LT(R)$$

and

$$LT(L(R))/\ker(h^*) \cong LT(R) .$$

Q.E.D.

3.14 is the algebraic alter ego of the statement that R is an extension of L(R). R is obtained from L(R) by augmenting L(R) with the theorems stating the relative recursiveness of functions.

We can now extend the theorems 2.5 and 2.6 concerning the linguistic suitability of R for recursion theory to theorems demonstrating the algebraic suitability of LT(R) for the study of the algebraic relations among degrees.

From our preliminary review of the theory of degrees we recall that $(D, \cup)$ (or $(D, \leq)$) is an upper semilattice. It is clear that $(D, \cup)$ is a substructure of LT(R) because LT(R) reflects the join of two functions as conjunction and the ordering among functions as the ordering of the algebra.

<u>Theorem 3.15</u>. $(D, \cup)$ (or $(D, \leq)$) is anti-isomorphic to a sub-semilattice of $LT(R)$.

<u>Proof</u>. Consider the mapping h from D into $LT(R)$.

$$h(a) = |\dot{}_{i\varepsilon I}A_i|$$

where a is the degree of f_k and $A_i = (\underline{f}_k(\underline{i}) = \underline{f_k(i)})$. $f_j \leq f_k \leq f_j$ iff $f_j \varepsilon a$. By 2.5 then

$$\dot{}_{i\varepsilon I}A_i \leftrightarrow \dot{}_{i\varepsilon I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)})$$

and by 3.8

$$|\dot{}_{i\varepsilon I}A_i| = |\dot{}_{i\varepsilon I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)})| .$$

Thus h is well defined and one-one. 2.5 and 3.8 show that if for $a,b \varepsilon D$ and $a \leq b$ then

$$h(a) \geq h(b),$$

and h is antitone.

Further, by 2.6, for functions f_j, f_k, f_m if

$$\mathrm{join}(f_j, f_k) = f_m$$

then

$$(\dot{}_{i\varepsilon I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)}))\cdot(\dot{}_{i\varepsilon I}(\underline{f}_k(\underline{i}) = \underline{f_k(i)})) \leftrightarrow \dot{}_{i\varepsilon I}(\underline{f}_m(\underline{i}) = \underline{f_m(i)})$$

is a theorem of R. Therefore, by 3.8,

$$|(\dot{\prod}_{i\in I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)}))|\cdot|(\dot{\prod}_{i\in I}(\underline{f}_k(i) = \underline{f_k(i)}))| =$$

$$|\dot{\prod}_{i\in I}(\underline{f}_m(\underline{i}) = \underline{f_m(i)})| .$$

Thus if $f_j \varepsilon a$ and $f_k \varepsilon b$, $f_m \varepsilon c$ for $a,b,c \varepsilon D$ then

$$h(a \cup b) = h(c) = |\dot{\prod}_{i\in I}(\underline{f}_m(\underline{i}) = \underline{f_m(i)})| =$$

$$|(\dot{\prod}_{i\in I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)}))\cdot(\dot{\prod}_{i\in I}(\underline{f}_k(\underline{i}) = \underline{f_k(i)}))| =$$

$$|\dot{\prod}_{i\in I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)})|\cdot|\dot{\prod}_{i\in I}(\underline{f}_k(\underline{i}) = \underline{f_k(i)})| =$$

$$h(a)\cdot h(b)$$

and h translates $\cup$ in $(D, \cup)$ to $\cdot$ in LT(R) and $\leq$ in $(D, \cup)$ to $\geq$ in LT(R). Clearly, the range of h is closed under $\cdot$.

Q.E.D.

Henceforth, we shall identify $(D, \leq)$ or $(D, \cup)$ with the range of the morphism h in 3.15. Notice that the structure $(D, \leq, \theta)$ (or $(D, \cup, \theta)$) is not preserved by h where θ, the degree of recursive functions, is the zero in the semilattice D. This is essentially because the zero is not treated as a structural feature. Indeed, the roles of zero in D and in LT(R), given the construction of LT(R), are incompatible. In LT(R) zero represents no degree so to attempt to preserve the zero of D would conflict with the constraints of 2.5 and 2.6. Notice also that h is antitone and that semilattice join is transferred to Boolean meet. We could avoid this incongruity if the theory of degrees were modified by reversing the orientation of the ordering of degrees.

Theorem 3.16. $a, b \in D$ and $f_j \in a$ and $f_k \in b$ then

$$a \leq b \text{ iff } \left|{}^{\cdot}_{i\in I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)})\right| \leq \left|{}^{\cdot}_{i\in I}(\underline{f}_k(\underline{i}) = \underline{f_k(i)})\right| .$$

Proof. Recall that D is now the range of h in 3.15. The theorem merely restates this fact.

Q.E.D.

The following theorem accomplishes the same purpose for join.

Theorem 3.17. $a, b, c \in D$ and $f_j \in a$, $f_k \in b$ and $f_m \in c$ then

$$a \cup b = c \text{ iff} \left|{}^{\cdot}_{i\in I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)})\right| \cdot \left|{}^{\cdot}_{i\in I}(\underline{f}_k(\underline{i}) = \underline{f_k(i)})\right| =$$

$$\left|{}^{\cdot}_{i\in I}\underline{f}_m(\underline{i}) = \underline{f_m(i)})\right| .$$

Proof. Same as 3.16.

Q.E.D.

We have established the principal connections of LT(R) to R and to the semilattice of degrees.

One of the aims of studying a theory through its Lindenbaum-Tarski algebra is to abstract from the linguistic vagaries of apparently different formulations of the theory. Our motivation is somewhat different; however, we do wish to eliminate inessential aspects of the theory through the algebraization. Rule ix), although superfluous to the formulation of a propositional recursion theory serves this purpose. We make further simplifications. LT(R) has elements inherited from R which, while necessary because of the formulation of R, tell us nothing about the algebraic relations among degrees. These elements of LT(R) are the vestiges of the apparatus of R which permit one function to be implied by another. They are the algebraic elements formed from equations involving auxiliary function symbols. From the logical point of view of R we know that for functions f_k and f_j

$$\dot{}_{i\epsilon I}(\underline{f}_k(\underline{i}) = \underline{f_k(i)}) \rightarrow \dot{}_{i\epsilon I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)})$$

is a theorem only when there exists some conjunction of auxiliary equations such that rule viii) applies. But from the algebraic viewpoint, it only matters that

$$|\dot{}_{i\epsilon I}(\underline{f}_k(\underline{i}) = \underline{f_k(i)})| \leq |\dot{}_{i\epsilon I}(\underline{f}_j(\underline{i}) = \underline{f_j(i)})| ,$$

not that such and such auxiliary equations must be used to establish the original implication in R. We will therefore be concerned with a subalgebra of LT(R) retaining all of the interesting features of LT(R) but none of the reminders of the origin of LT(R) in a deductive system. The subalgebra, A(R), is the G-subalgebra generated by $\{|A| \mid A$ is a numeric equation$\}$. Where G is the collection of all sets of the form $\{|\underline{f}_j(\underline{i}) = \underline{f_j(i)}| \mid i\epsilon I\}$ for $j\epsilon I_1$. A(R) has some pleasing algebraic regularities based upon the fundamental ordering of degrees.

The Algebra A(R)

The following theorem sets down two evident properties of A(R).

Theorem 3.18. The G-algebra A(R) generated by $\{|A| \mid A$ is a numeric equation in R$\}$ is a regular subalgebra of LT(R).

<u>Proof</u>. A(R) is obviously a subalgebra of LT(R). The regularity of A(R) is assured by the observation that the generators of A(R) are a subset of the generators of LT(R), thus

$$\dot{\prod_{j\in J}} \overset{+}{\sum_{k\in K}} d(j,k)|A_{j,k}| \text{ and } \dot{\prod_{i\in I}}|A_i|$$

where A_i and $A_{j,k}$ are numeric equations, are identical in A(R) and LT(R).

Q.E.D.

Notice that A(R) retains the features which made LT(R) interesting for the study of the algebra of degrees. Theorems 3.16 and 3.17 obviously apply to A(R) since D is a substructure of A(R). A(R) can be analyzed extensively in terms of D, or rather, in terms of the subalgebra generated by D in A(R).

CD is the algebra <u>generated by D</u> in A(R). Because D is the set of all infinite infima closed under · in A(R), CD is the subalgebra generated by the set of all infinite infima in A(R). We have the following theorem.

<u>Theorem 3.19</u>. CD is a regular subalgebra of A(R) and CD is the Boolean completion of D.

<u>Proof</u>. By definition CD is a subalgebra of A(R). That CD is generated by D and D is regular in CD is clear. The regularity of CD is obvious.

Q.E.D.

Recall that the <u>zero of D</u>, which we call θ, is not considered a structural feature of D. Therefore, the Boolean completion of D, CD, does not treat θ distinctly from the other elements. CD is the Boolean algebra achieved by adding to D complements and finite suprema. We shall see that CD, simple as it is, reveals everything about the structure of A(R).

For each $i \epsilon I$, let $\underline{A(R)_i}$ be the subalgebra generated by $\{|A_i| \mid A_i = (\underline{f}_j(\underline{i}) = \underline{f_j(i)}), j \epsilon I_1\}$. $A(R)_i$ is the subalgebra of A(R) which has as generators all the i^{th} elements of the infinite infima of A(R). Notice that the selection of the i^{th} element of the infimum $\dot{\underset{i \epsilon I}{}}|A_i|$ is not dependent upon any ordering of the elements of the infimum. The i^{th} element is the element representative of equations involving the value of the function at the i^{th} n-tuple in the ordering of n-tuples which we assumed. Thus, $|\underline{f}_j(\underline{i}) = \underline{f_j(i)}|$ is the i^{th} element of $\dot{\underset{i \epsilon I}{}}|\underline{f}_j(\underline{i}) = \underline{f_j(i)}|$ regardless of any incidental ordering of the infimum. Each subalgebra $A(R)_i$ is isomorphic to CD.

<u>Theorem 3.20</u>. $A(R)_i \cong CD$

<u>Proof</u>. This theorem is the result of rule ix) of R. Consider the mapping h of the generators of $A(R)_i$ onto the generators of CD defined

$$h(|A_i|) = \dot{\underset{i \epsilon I}{}}|A_i| \ .$$

Let $|G_{j,i}| = |\underline{f}_{k_j}(\underline{i}) = \underline{f_{k_j}(i)}| = |A_i|$. By 2.1 h can be extended to an isomorphism if and only if for $j \leq m$, and $d(j,i) = (+)$ or $(-)$

$$\dot{\underset{j \leq m}{}} d(j,i)|G_{j,i}| = 0 \quad \text{iff}$$

$$\overset{\cdot}{}_{j\leq m}d(j,i)h(|G_{j,i}|) = 0 \ .$$

By the definition of h,

$$\overset{\cdot}{}_{j\leq m}d(j,i)h(|G_{j,i}|) = \overset{\cdot}{}_{j\leq m}d(j,i)(\overset{\cdot}{}_{i\varepsilon I}|G_{j,i}|) \ .$$

We wish to use distributive laws for which the indices i must be notationally distinct; therefore, we subscript the i with j and the right hand side of the above equation is equal to

$$(1) \qquad \overset{\cdot}{}_{j\leq m}d(j,i_j)(\overset{\cdot}{}_{i_j\varepsilon I}|G_{j,i_j}|)$$

Where $X(j,i_j)$ is $\overset{\cdot}{}_{i_j\varepsilon I}$ if $d(j,i_j) = (+)$, $X(j,i_j)$ is $\overset{+}{}_{i_j\varepsilon I}$ if $d(j,i_j) = (-)$, and by repeated applications of distributive laws the above element is equal to

$$X(1,i_1)X(2,i_2)\ldots X(m,i_m)\overset{\cdot}{}_{j\leq m}d(j,i_j)|G_{j,i_j}|$$

which by 3.8 and 3.10 equals

$$|X(1,i_1)X(2,i_2)\ldots X(m,i_m)\overset{\cdot}{}_{j\leq m}d(j,i_j)G_{j,i_j}|$$

where $X(j,i_j)$ is now reinterpreted as infinite conjunction or disjunction.

But if

$$|\overset{\cdot}{}_{j\leq m}d(j,i)G_{j,i}| = 0$$

then

$$|\overset{\cdot}{}_{j\leq m}d(j,i_j)G_{j,i_j}| = 0$$

since there is only a notational variation. Thus, by 3.8,

$$\dot{\sum_{j\leq m}} d(j,i_j)G_{j,i_j} \leftrightarrow (A\cdot(-A))$$

is a theorem of R.

Therefore,

$$X(1,i_1)X(2,i_2)\ldots X(m,i_m)\dot{\sum_{j\leq m}} d(j,i_j)G_{j,i_j} \leftrightarrow (A\cdot(-A))$$

is a theorem of R regardless of the nature of the infima and suprema $X(j,i_j)$. So, by 3.8,

$$|X(1,i_1)X(2,i_2)\ldots X(m,i_m)\dot{\sum_{j\leq m}} d(j,i_j)G_{j,i_j}| = 0 .$$

But the left hand side of this equation is equal to (1), thus

$$\dot{\sum_{j\leq m}} d(j,i)h(|G_{j,i}|) = 0 .$$

On the other hand, if $\dot{\sum_{j\leq m}} d(j,i)h(|G_{j,i}|) = 0$ then by an argument similar to the foregoing

$$X(1,i_1)X(2,i_2)\ldots X(m,i_m)\dot{\sum_{j\leq m}} d(j,i_j)G_{j,i_j} \leftrightarrow (A\cdot(-A))$$

is a theorem of R. But this is only possible if

$$\dot{\sum_{j\leq m}} d(j,i_j)G_{j,i_j} \leftrightarrow (A\cdot(-A))$$

is a theorem of R. Thus by 3.8,

$$|\dot{\sum_{j\leq m}} d(j,i_j)G_{j,i_j}| = 0$$

and

$$\dot{\sum_{j\leq m}} d(j,i_j)|G_{j,i_j}| = 0$$

Dropping the subscript on the i, we have

$$\dot{\sum_{j\leq m}} d(j,i)\,|G_{j,i}| = 0 \ .$$

Therefore, we have established the biconditional and we may extend h to an isomorphism.

Q.E.D.

A(R) can in a sense be "decomposed" into an algebra generated by the Boolean product of the $A(R)_i$ and CD. This is a fundamental structural feature of A(R) in light of 3.20. One way of looking at the structure of A(R) is first to remove from A(R) all elements involving infinite infima or suprema. What remains is the Boolean product of the $A(R)_i$. Now if we consider the algebra generated by the union of the Boolean product of the $A(R)_i$ with CD we see that we have merely re-introduced those infinite infima and suprema. This notion receives its precise statement in the following theorems.

Notice that the algebras $A(R)_i$ are independent since their generators are.

<u>Theorem 3.21</u>. $B_{i\in I}A(R)_i$ is a subalgebra of A(R).

<u>Proof</u>. For each $i\in I$, $A(R)_i$ is a subalgebra of A(R) by definition. Thus, $B_{i\in I}A(R)_i$, which is the subalgebra generated by the union of the $A(R)_i$, is a subalgebra of A(R).

Q.E.D.

We know by the definition of CD that it is a subalgebra of A(R), thus the following.

<u>Theorem 3.22</u>. $\{B_{i\in I}A(R)_i \cup CD\} = A(R)$.

<u>Proof</u>. From 3.21 and the fact that CD is a subalgebra of A(R), we know $\{B_{i\in I}A(R)_i \cup CD\}$ is a subalgebra of A(R). For any element $A\in A(R)$ we can obviously write

$$A = \mathop{+}_{j\leq m} \mathop{\cdot}_{k\leq n} d(j,k)G_{j,k}$$

where for a given j,k, $G_{j,k}$ is either a generator of A(R) or $G_{j,k}$ is an infinite infimum. Then clearly $G_{j,k}\in B_{i\in I}A(R)_i$ or $G_{j,k}\in CD$. In either case, $A\in\{B_{i\in I}A(R)_i \cup CD\}$, thus

$$\{B_{i\in I}A(R)_i \cup CD\} = A(R).$$

Q.E.D.

3.22 confirms our earlier statement that CD "tells everything" about the structure of A(R). The decomposition of A(R) in terms of CD will prove particularly useful in the construction of abstract digital computers for the various degrees.

From this point as a notational convenience we shall rely upon the identification of $(D, \leq)$ with a substructure of A(R) and we will indicate degrees in A(R) by lower-case roman letters, a,b,c,.... In particular we shall identify θ with $|\mathop{\cdot}_{i\in I}A_i|$ where $A_i = (\underline{f}_k(\underline{i}) = \underline{f_k(i)})$ and f_k is a recursive function. Therefore, θ is the degree of recursive functions and for any degree $b\in D$, $b\leq\theta$.

We now consider some properties of CD, in particular, its relation to the ideal $(\theta]$.

A(R) quotiented by the principal ideal generated by a degree will be considered to be the algebraic structure of that degree. Thus for $b \varepsilon D$, $A(R)/(b]$ is the structure of the degree b. The reason we take this algebra to represent a degree will become fully evident later, but there are some available intuitions at this point. A(R) carries all the information about the algebraic relations among degrees and $(b]$ contains every degree smaller than b. Thus if $c \leq b$ then $(c] \subseteq (b]$ and $A(R)/(b]$ is isomorphic to a quotient of $A(R)/(c]$. We may then say that if $c \leq b$ then the structure of b is isomorphic to a quotient of the structure of c. The ordering of degrees is then translated into an algebraic relation among the algebras representing the degrees. Such a translation is necessary if we are to recover the ordering among degrees as an algebraic relation among abstract digital computers representing degrees. But we must establish these heuristic remarks.

Theorem 3.23. If $(a] \subseteq (b]$ then $A(R)/(b]$ is isomorphic to a quotient of $A(R)/(a]$.

Proof. Let $|c|_a$ and $|c|_b$ designate the representatives of $|c|$ in $A(R)/(a]$ and $A(R)/(b]$ respectively. Define h from $A(R)/(a]$ onto $A(R)/(b]$ as

$$h(|c|_a) = |c|_b .$$

h is clearly an onto function since if $|c|_a = |d|_a$ then

$$|c|_b = |d|_b \ .$$

h is also a homomorphism.

$$h(|c|_a \cdot |d|_a) = h(|c \cdot d|_a) = |c \cdot d|_b =$$
$$|c|_b \cdot |d|_b = h(|c|_a) \cdot h(|d|_a)$$

and

$$h(-|c|_a) = h(|-c|_a) = |-c|_b =$$
$$-|c|_b = -h(|c|_a) \ .$$

Therefore

$$A(R)/(b] \cong A(R)/(a]/\ker(h) \ .$$

Q.E.D.

The next theorem shows that the decomposition of A(R) in terms of CD holds, in a modified form, for A(R)/(d].

Theorem 3.24. $A(R)/(d] \cong \{B_{i \in I} A(R)_i \cup CD/(d]\}$.

Proof. Notice that (d] in A(R) is distinct from (d] in CD because the only elements of CD smaller than d are generated from the infinite infima, but the elements of A(R) smaller than d are generated by infinite infima and generators of A(R).

If A is a generator of A(R), then $|A|_d$ is a singleton. Suppose, to the contrary, that for generators A and B, $A \neq B$ and $|A|_d = |B|_d$. Then $((-A) \cdot B) + (A \cdot (-B)) \in (d]$ and $((-A) \cdot B), (A \cdot (-B)) \in (d]$. But

$A = |\underline{f}_j(\underline{i}) = \underline{f_j(i)}|_{A(R)}$ and $B = |\underline{f}_k(\underline{n}) = \underline{f_k(n)}|_{A(R)}$ for some functions f_j and f_k, and $d = |\dot{\prod}_{i\varepsilon I}(\underline{f}_m(\underline{i}) = (\underline{f_m(i)})|$ for some f_m. If

$((-A)\cdot B)\varepsilon(d]$ then $((-A)\cdot B) \leq d$; consequently,

$$(1) \qquad (-(\underline{f}_j(\underline{i}) = \underline{f_j(i)})\cdot(\underline{f}_k(\underline{n}) = \underline{f_k(n)})) \rightarrow \dot{\prod}_{i\varepsilon I}(\underline{f}_m(\underline{i}) = \underline{f_m(i)})$$

is a theorem in R and this is clearly impossible where k, j, and m are distinct. (1) does not have the form of any theorem of R.

The set of generators of A(R) together with the set of infinite infima of A(R) constitute a set of generators for A(R). Therefore, for $A\varepsilon A(R)$

$$A = \overset{+}{\sum}_{j\leq n} \dot{\prod}_{k\leq m} d(j,k)G_{j,k}$$

where $d(j,k) = (+)$ or $(-)$ and for each $j,k,G_{j,k}$ is either a generator of A(R) or an infinite infimum in A(R).

Now define h from $A(R)/(d]$ to $|B_{i\varepsilon I}A(R)_i \cup CD/(d]|$ as follows:

$$(2) \qquad h(|A|_d) = h(|\overset{+}{\sum}_{j\leq n} \dot{\prod}_{k\leq m} d(j,k)G_{j,k}|_d) =$$

$$\overset{+}{\sum}_{j\leq n} \dot{\prod}_{k\leq m} d(j,k)|G_{j,k}|_{d^*}$$

where $|G_{j,k}|_{d^*}$ is $|G_{j,k}|_d$ in CD if $G_{j,k}\varepsilon CD$ and $|G_{j,k}|_{d^*}$ is $G_{j,k}$ if $G_{j,k}\varepsilon A(R)_i$ for some $i\varepsilon I$. If h is a one-one onto function, then it is, by (2), an isomorphism.

Suppose $|A|_d = |C|_d$ and

$$A = \overset{+}{\sum}_{j'\leq n'} \dot{\prod}_{k'\leq m'} d(j',k')G'_{j',k'}$$

$$C = \overset{+}{\underset{j\leq n}{}}\ \overset{\cdot}{\underset{k\leq m}{}} d(j,k)G_{j,k}$$

then

$$|_{j'\leq n'}\overset{+}{}\ {}_{k'\leq m'}\overset{\cdot}{} d(j',k')G'_{j',k'}|_d = |_{j\leq n}\overset{+}{}\ {}_{k\leq m}\overset{\cdot}{} d(j,k)G_{j,k}|_d$$

and by repeated applications of 3.8,

$$\overset{+}{\underset{j'\leq n'}{}}\ \overset{\cdot}{\underset{k'\leq m'}{}} d(j',k')|G'_{j',k'}|_d = \overset{+}{\underset{j\leq n}{}}\ \overset{\cdot}{\underset{k\leq m}{}} d(j,k)|G_{j,k}|_d .$$

If $G'_{j',k'}$ or $G_{j,k}$ is a generator then

$$|G'_{j',k'}|_{d*} = G'_{j',k'} \quad \text{and} \quad |G_{j,k}|_{d*} = G_{j,k} \quad \text{and } *$$

is a one-one function since $|G'_{j',k'}|_d$ and $|G_{j,k}|_d$ are singletons. If $G'_{j',k'}$ or $G_{j,k}$ are infinite infima then

$$|G'_{j',k'}|_{d*} = |G'_{j',k'}|_d \text{ and } |G_{j,k}|_{d*} = |G_{j,k}|_d$$

where the right hand sides are in CD. But CD is a regular subalgebra of A(R), thus $|G_{j,k}|_d$ in A(R) is equal to $|G_{j,k}|_d$ in CD and * is still a one-one function

Therefore,

$$h(|A|_d) = \overset{+}{\underset{j'\leq n'}{}}\ \overset{\cdot}{\underset{k'\leq m'}{}} d(j',k')|G'_{j',k'}|_{d*} =$$

$$\overset{+}{\underset{j\leq n}{}}\ \overset{\cdot}{\underset{k\leq m}{}} d(j,k)|G_{j,k}|_{d*} = h(|B|_d)$$

and h is a one-one function.

h is onto because for $G_{j,k} \varepsilon B_{i\varepsilon I} A(R)_i \cup CD/(d]$, by (2) above

$$\overset{+}{\underset{j\le n}{}}\ \overset{\cdot}{\underset{k\le m}{}} d(j,k)G_{j,k} = \overset{+}{\underset{j\le n}{}}\ \overset{\cdot}{\underset{k\le m}{}} d(j,k)|G_{j,k}|_{d^*} =$$

$$h(|\overset{+}{\underset{j\le n}{}}\ \overset{\cdot}{\underset{k\le m}{}} d(j,k)G_{j,k}|)$$

Therefore, h is the desired isomorphism.

Q.E.D.

Theorem 3.25. θ is a dual atom of CD.

Proof. Suppose $\overset{+}{\underset{i\le n}{}}\ \overset{\cdot}{\underset{j\le m}{}} d(i,j)G_{i,j} \geq \theta$, where $G_{i,j}$ are generators of CD and m is dependent on i. Recall that the generators of CD are degrees. Then we have two cases:

i) If for each $i \leq n$ there exists $j \leq m$ such that

$$d(i,j) = (+) ,$$

then

$$d(i,j)G_{i,j} \leq \theta$$

thus

$$\overset{\cdot}{\underset{j\le m}{}} d(i,j)G_{i,j} \leq \theta .$$

ii) If for some $i \leq n$, $d(i,j) = (-)$ for all $j \leq m$ then, since

$$-G_{i,j} \geq -\theta$$

we have

(1) $$\overset{\cdot}{\underset{j\le m}{}} -G_{i,j} \geq -\theta .$$

Without loss of generality, we assume the disjuncts greater than $-\theta$ to be indexed by 1 through $k \leq n$. Then, since each $G_{i,j} \leq \theta$

(2) $$\theta \geq \mathop{+}\limits_{k+1 \leq i \leq n} \mathop{\cdot}\limits_{j \leq m} d(i,j)G_{i,j}$$

and from (1)

(3) $$-\theta \leq \mathop{+}\limits_{i \leq k} \mathop{\cdot}\limits_{j \leq m} -G_{i,j}$$

but by the hypothesis, (2) and (3)

$$\mathop{+}\limits_{i \leq n} \mathop{\cdot}\limits_{j \leq m} d(i,j)G_{i,j} = (\mathop{+}\limits_{k+1 \leq i \leq n} \mathop{\cdot}\limits_{j \leq m} d(i,j)G_{i,j}) + (\mathop{+}\limits_{i \leq k} \mathop{\cdot}\limits_{j \leq m} -G_{i,j}) =$$

$$\theta + (\mathop{+}\limits_{k+1 \leq i \leq n} \mathop{\cdot}\limits_{j \leq m} d(i,j)G_{i,j}) + (\mathop{+}\limits_{i \leq k} \mathop{\cdot}\limits_{j \leq m} -G_{i,j}) =$$

$$\theta + (\mathop{+}\limits_{i \leq k} \mathop{\cdot}\limits_{j \leq n} -G_{i,j}) \geq \theta + (-\theta) = 1$$

and θ is a dual atom as stated.

Q.E.D.

Theorem 3.26. $-\theta$ is an atom of CD.

Proof. Immediate dualization of 3.25.

Q.E.D.

Theorem 3.27. $CD/(\theta] = 2$.

Proof. The principal ideal of a dual atom is maximal (Sikorski [37], p. 28) and the quotient of an algebra by a maximal ideal is isomorphic to 2 (Sikorski [37], p. 32).

Q.E.D.

We prove an interesting corollary to 3.24.

<u>Theorem 3.28</u>. $A(R)/(\theta] = B_{i\epsilon I}A(R)_i$.

<u>Proof</u>. By 3.27 $CD/(\theta] = 2$, and by 3.24

$$A(R)/(\theta] = \{B_{i\epsilon I}A(R)_i \cup CD/(\theta]\}$$

thus

$$A(R)/(\theta] = \{B_{i\epsilon I}A(R)_i \cup 2\}$$

But

$$B_{i\epsilon I}A(R)_i \cup 2 = B_{i\epsilon I}A(R)_i$$

since 2 is a subalgebra of $B_{i\epsilon I}A(R)_i$; therefore

$$A(R)/(\theta] = \{B_{i\epsilon I}A(R)_i\}$$

and $B_{i\epsilon I}A(R)_i$ is the subalgebra generated by the $A(R)_i$; therefore

$$A(R)/(\theta] = B_{i\epsilon I}A(R)_i \ .$$

Q.E.D.

The following theorem establishes the relationship between the structures of degrees and their joins.

<u>Theorem 3.29</u>. For $a,b,c,d\epsilon D$ if $a\cdot b = c$ and $d \leq a$ and $d \leq b$ then if $A(R)/(a]$ is isomorphic to a quotient of $A(R)/(d]$ and so is $A(R)/(b]$ then $A(R)/(c]$ is isomorphic to a quotient of $A(R)/(d]$.

<u>Proof</u>. Obviously if $d \leq a$ and $d \leq b$ then $d \leq a\cdot b = c$. Thus $(d] \subseteq (a\cdot b] = (c]$ and by 3.23 $A(R)/(c]$ is isomorphic to a quotient of $A(R)/(d]$.

Q.E.D.

Abstract Digital Computers for Degrees

The algebraic considerations of LT(L(R)), LT(R), A(R) and CD are for the purpose of reconstructing the elementary theory of degrees within the theory of abstract digital computers. To that purpose we first must have a Boolean algebra in which the ordering of degrees and the join operation of degrees are adequately represented. LT(R) meets these criteria by 3.16 and 3.15. Theorems 3.16 and 3.15 also apply to A(R), and we find A(R) to be a convenient algebra in which to work in view of 3.23 and 3.29.

The decomposition of A(R) via 3.22 and of A(R)/(d] via 3.24 affords a simple construction of abstract digital computers for degrees. Notice that in A(R)/(d] the subalgebra $B_{i\varepsilon I}A(R)_i$ remains as it is in A(R) up to the identification of elements with their singleton sets. We take advantage of this fact to construct a transition function for the degree d that is independent of A(R)/(d] but dependent upon the degree d. We give the construction, then prove the appropriate theorems.

Recall that from 3.20 $A(R)_i \cong$ CD. Thus for each degree dεCD there is a <u>corresponding element in $A(R)_i$</u>, for every iεI call it d_i. Consider the functions of d to be well ordered. We know that there are only I functions in d; thus we may assign each function to a distinct subprogression of $(d_i|i\varepsilon I)$ such that the progressions are mutually disjoint and each isomorphic to I. For $f_j \varepsilon d$ call df_j the <u>progression assigned to</u> f_j and let $(df_j)_k$ be the <u>k^{th} element in the progression</u>. Now repeat this procedure for every cεD. We require that for any d,cεD, the progressions assigned to the functions of d and c be uniformly chosen. If the least (in the well ordering) function $f_k \varepsilon d$ is assigned the progression $(df_k)_1 = d_{i_1}$,

$(df_k)_2 = d_{i_2}$, $(df_k)_3 = d_{i_3}, \ldots$ then the least function $f_n \epsilon c$ is assigned the progression $(cf_n)_1 = c_{i_1}$, $(cf_n)_2 = c_{i_2}$, $(cf_n)_3 = c_{i_3}, \ldots$. Then for the n^{th} function of d and n^{th} function of c, the progression assigned to each correspond in the indices; that is, the m^{th} element in each progression has the same index. Then define T_c as follows.

$$T_{cf_j}(A) = \begin{cases} (cf_j)_m & \text{if } A = (cf_j)_k \text{ and } f_j(k) = m \\ \text{undefined, otherwise} \end{cases}$$

$$T_c(A) = \begin{cases} T_{cf_j}(A) & \text{if } A = (cf_j)_k, \text{ for some } k \\ \text{undefined, otherwise.} \end{cases}$$

Now define T_d^* on $A(R)/(d]$ as follows for $A \epsilon A(R)/(d]$

$$T_d^*(A) = \begin{cases} T_c(A) & \text{if } T_c(A) \text{ is defined and } d \leq c. \\ \text{undefined, otherwise.} \end{cases}$$

Notice that T_d^* is well defined since the T_c's are all disjoint and that the T_c's are well defined since the T_{cf_j}'s are disjoint. Further notice that T_d^* is undefined on every element of $A(R)_i$ less than d_i, and that T_d^* is undefined entirely outside of $A(R)_i$ for $i \epsilon I$.

<u>Theorem 3.30</u>. $(A(R)/(d], T_d^*)$ is an abstract digital computer.

<u>Proof</u>. Obvious.

Notice that if h is an embedding of $(A(R)/(d], T_d^*)$ into $(A(R)/(c], T_c^*)$ then if for $f_n \epsilon a \leq d$ and $f_p \epsilon b \leq c$

$$h(T_d^*((af_n)_m)) = T_c((bf_p)_s)$$

then

$$T_c^*(h((af_n)_m)) = T_c^*((bf_p)_s)$$

and

$$h((af_n)_m) = (bf_p)_s$$

Thus h must map the process given by T_d^* started on $(af_n)_m$ into the process given by T_c^* started on $(bf_p)_s$. Therefore h restricted to the domain process is a partial function from the *progression* (af_n) into the *progression* (bf_p). Each of these progressions is naturally isomorphic to I. Thus h as restricted may be viewed as a partial function of I into I. We can now make the following definitions.

If h maps (df_n) into (cf_p), such that when taken as a mapping on the *indices*, h is recursive, then we say that h is *r-recursive* on (df_n) into (cf_p). An embedding h of $(A(R)/(d], T_d^*)$ into $(A(R)/(c], T_c^*)$ is an *r-embedding* iff h as restricted to each of the domain progressions is r-recursive on each progression.

Although we do not know in general whether or not $A(R)/(d] \cong A(R)/(c]$ for distinct degrees d and c, we have the following.

Theorem 3.31. If $a \not\leq d$ then $(A(R)/(d], T_d^*)$ is not r-embeddable in a quotient of $(A(R)/(a], T_a^*)$.

Proof. Let $a \not\leq d$. Assume h of $(A(R)/(d], T_d^*)$ into (B, T_B) to be a r-embedding where (B, T_B) is a quotient of $(A(R)/(a], T_a^*)$. Notice that if

$$T_a^* = T_{cf}$$

on the progression (cf), then for $m,n \in I$ if

$$T_a^*((cf)_m) = (cf)_n \text{ then } T_B(|(cf)_m|_B) =$$

$$|T_a^*((cf)_m)|_B = |(cf)_n|_B$$

Therefore, for $f_n \in d$, if

$$T_d^*((df_n)_m) = T_{df_n}((df_n)_m) = (df_n)_{f_n(m)},$$

there must exist some $c \geq a$ and $f_p \in c$ such that

$$h(T_d^*((df_n)_m)) = T_B(h((df_n)_m)) = T_B(|(cf_p)_s|_B),$$

for some $s \in I$.

Also,

$$T_B(|(cf_p)_s|_B) = |T_a^*((cf_p)_s|_B = |T_{cf_p}((cf_p)_s)|_B =$$

$$|(cf_p)_{f_p(s)}|_B .$$

We clearly have that

$$T_{cf_p}((cf_p)_s) = (cf_p)_t \text{ iff } f_p(s) = t$$

thus

$$T_B(|(cf_p)_s|_B) = |(cf_p)_t|_B \text{ iff } f_p(s) = t .$$

Also clearly,

$$T_{df_n}((df_n)_r) = (df_n)_u \text{ iff } f_n(r) = u .$$

Thus if we consider the natural isomorphisms of

(1) $$|(cf_p)_1|_B,\ |(cf_p)_2|_B, \ldots, |(cf_p)_n|_B, \ldots$$

and (df_n) into I. T_B restricted to the progression (1), indicated T_{Bcf_p}, is co-recursive with f_p and T_{df_n} is co-recursive with f_n. Thus,

$$h(T_{df_n}((df_n)_s)) = T_{Bcf_p}(h((df_n)_s))$$

and

$$h^{-1}(T_{Bcf_p}(|(cf_p)_m|_B)) = T_{df_n}(h^{-1}(|(cf_p)_m|_B))$$

because h is a computer embedding.

Then,

$$T_{df_n}((df_n)_s) = h^{-1}(T_{Bcf_p}(h((df_n)_s)))$$

and

$$T_{Bcf_p}(|(cf_p)_m|_B) = h(T_{df_n}(h^{-1}(|(cf_p)_m|_B))) .$$

But h and h^{-1} on the progressions (df_n) and (1) are recursive; thus

T_{df_n} and T_{Bcf_p} are co-recursive.

Therefore, $f_n \varepsilon d$ is co-recursive to $f_p \varepsilon c$ and

$$d = c$$

contrary to the hypothesis since $c \geq a$.

Q.E.D.

We conclude the recovery of the elementary theory of degrees in the theory of abstract digital computers with the analogs to theorems 3.23 and 3.29.

<u>Theorem 3.32</u>. If $a \leq b$ then $(A(R)/(b], T_b^*)$ is r-embeddable in a quotient of $(A(R)/(a], T_a^*)$.

<u>Proof</u>. From 3.23 $A(R)/(b]$ is isomorphic to a quotient of $A(R)/(a]$. $A(R)/(b] = \{B_{i\epsilon I}A(R)_i \cup CD/(b]\}$ and $A(R)/(a] = \{B_{i\epsilon I}A(R)_i \cup CD/(a]\}$ by 3.24. The transition functions T_b^* and T_a^* are undefined outside of $B_{i\epsilon I}A(R)_i$, but they coincide wherever T_b^* is defined because for every $f\epsilon c \geq b$, $f\epsilon c \geq a$ and

$$T_b^* = T_{cf} = T_a^*$$

on the progression (cf). Therefore

$$q: A(R)/(b] \cong A(R)/(a]/\ker(h)$$

where h is the homomorphism of 3.23, can preserve the transition functions wherever T_b^* is defined.
Notice that

$$A(R)/(b]/\ker(h) \cong \{B_{i\epsilon I}A(R)_i, CD/(a]/\ker(h)\}$$

by 3.23 and 3.24. So

$$q(T_b^*(A)) = T_a^*(q(A))$$

if T_b^* is defined on A, and we see q sends T_b^* to T_a^*.

Q.E.D.

The next theorem of this section is the counterpart to 3.29.

Theorem 3.33. For $a,b,c,d \in D$ if $a \cdot b = c$ and $d \leq a$ and $d \leq b$, $(A(R)/(c],T_c^*)$ is r-embeddable in a quotient of $(A(R)/(d],T_d^*)$.

Proof. $d \leq a \cdot b = c$ so the theorem follows by 3.32.

Q.E.D.

The final theorem demonstrates that the set of abstract digital computers of degrees has the structure of the degrees.

Theorem 3.34. The set of abstract digital computers for degrees under the converse of "r-embeddable in a quotient" is isomorphic to D under $\leq$.

Proof. By 3.32 if $a \leq b$ then $(A(R)/(b],T_b^*)$ is r-embeddable in a quotient of $(A(R)/(a],T_a^*)$. Theorem 3.33 gives the existence of the infimum. The assignment of degrees to computers is unique; therefore the isomorphism holds.

Q.E.D.

Concluding Remarks

The aim of this thesis is to reconstruct the elementary theory of degrees of unsolvability in the theory of abstract digital computers. In order to achieve this aim we show, by 3.30, that abstract digital computers for degrees exist. Further, 3.31 and 3.32 together show that the abstract digital computer of a degree, c, is r-embeddable in a quotient of the abstract digital computer of another degree, d, if and only if $d \leq c$. The capstone theorem, 3.34, shows that the set of abstract digital computers of degrees has the same structure under the relation "r-embeddable in a quotient of" as D has under $\leq$. A variety of continuations of this research suggest themselves.

The most pressing of these is the discovery of an independent characterization of the class of abstract digital computers of degrees. One might hope that this class is an equational class. Unfortunately, the class is not closed under the formation of subalgebras; therefore, it is not equational. However, a weaker characterization of the class may be available; for instance, we might find a general axiomatization of the class. Research in this direction is continuing.

In another direction the results of this thesis might be generalized if the concept "r-embeddable" can be exchanged for a "purely" algebraic concept. 3.31 takes its partially combinatorial nature from the concept of "r-embeddable." If a version of 3.31 substituting "embedding" for "r-embedding" is possible, then 3.31 may be improved by eliminating recursive embeddings in favor or ordinary embeddings. Once this is achieved, we might generalize the concept of degree to all abstract digital computers by partitioning the computers into "degrees" via the relation "embeddable in a quotient of." This partitioning is possible only if the relation is a partial ordering, which it appears not to be. While the relation is clearly transitive and reflexive, antisymmetry is a problem. One cannot even show in general that if a Boolean algebra A is isomorphic to a quotient of B and vice versa, then A and B are isomorphic. If the algebras are I-complete, "isomorphic to a quotient of" is a partial ordering (see [37], pp. 90, 193), but the Boolean algebras of abstract digital computers we are usually concerned with are not

I-complete, and their I-completions need not be distinct from one another. So, if the relation "isomorphic to a quotient of" is not a partial order, "embeddable in a quotient of" which contains it cannot be. Nonetheless, there may be an algebraic concept of degrees for abstract digital computers. A "recursive" theory of degrees for abstract digital computers is available.

Given a class of abstract digital computers, we can enumerate, in some ordinal, the elements of all the Boolean algebras of the class. Then the transition functions may be construed as functions on ordinals, and the theories of meta-recursion and meta-degrees [34] may be used to assess the degrees of the computers. Of course, the degree of a given computer is dependent upon the original enumeration chosen. A "purely" algebraic theory of degrees for abstract digital computers would, presumably, be invariant.

REFERENCES

1. Boone, William W. 1957. "Certain Simple, Unsolvable Problems of Group Theory, V, VI." Indagationes Mathematicae 19: 22-27.

2. Church, Alonzo. 1936. The Calculus of Lambda Conversion. Princeton: Princeton University Press.

3. ______________. 1936. "An Unsolvable Problem of Elementary Number Theory." American Journal of Mathematics 58: 345-363.

4. DeMillo, Richard. 1972. "Formal Semantics and the Logical Structure of Programming Languages," Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, Georgia.

5. Fefferman, Solomon. 1957. "Degrees of Unsolvability Associated with Classes of Formalized Theories." Journal of Symbolic Logic 22: 161-175.

6. Freidberg, R. M. 1957. "Two Recursively Enumerable Sets of Incomparable Degrees of Unsolvability." Proceedings of the National Academy of Sciences 43: 236-238.

7. Godel, Kurt. 1931. "On Formally Undecidable Propositions of the Principia Mathematica." Translated by Elliott Mendelson in The Undecidable, edited by Martin Davis. 1965. Hewlett, N.Y.: Raven Press.

8. Grzegorczyk, Andrzej. 1957. "Undecidability of some Topological Theories." Fundamentica Mathematica 38: 137-152.

9. Halmos, Paul. 1962. Algebraic Logic. New York: Chelsea.

10. Horgan, J., Roehrkasse, R., and Chiaraviglio, L. 1972. "Abstract Digital Computers and Turing Machines." Research Report GITIS-72-01, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Georgia.

11. Jonsson, B. and Tarski, A. 1951. "Boolean Algebras with Operators, I." Amer. J. of Math. 73: 891-939.

12. Jonsson, B. and Tarski, A. 1951. "Boolean Algebras with Operators, II." Amer. J. of Math. 74: 124-162.

13. Karp, Carol. 1964. Languages with Expressions of Infinite Length. Amsterdam: North Holland.

14. Kleene, Stephen C. 1936. "General Recursive Functions of Natural Numbers." Mathematische Annalen 112: 727-742.

15. ______________. 1936. "Lambda-Definability and Recursiveness." Duke Mathematical Journal 2: 340-353.

16. ______________. 1943. "Recursive Predicates and Quantifiers." Transactions of the American Mathematical Society 53: 41-73.

17. ______________. 1952. Introduction to Metamathematics. Princeton, N. J.: Van Nostrand.

18. Kleene, Stephen C. and Post, Emil L. 1954. "The Upper Semi-Lattice of Degrees of Recursive Unsolvability." Annals of Mathematics, ser. 2, 59: 379-407.

19. Mostowski, Andrzej. 1952. Sentences Undecidable in Formalized Arithmetic. Amsterdam: North Holland.

20. Mostowski, Andrzej and Tarski, Alfred. 1949. "Undecidability in the Arithmetic of the Integers and in the Theory of Rings." JSL 14: 76.

21. Muchnik, A. A. 1956. "On the Unsolvability of the Problem of Reducibility in the Theory of Algorithms." Doklady Akademii Nauk SSSR, n.s. 108: 194-197.

22. Novikov, P. S. 1955. "On the Algorithmic Unsolvability of the Word Problem in Group Theory." Trudy Mathematicheskoga Instituta imeni V. A. Steklova, no. 44.

23. Poore, Jesse H. 1970. "Toward an Algebra of Computation," Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, Georgia.

24. Post, Emil L. 1936. "Finite Combinatory Processes." JSL 1: 103-105.

25. ______________. 1943. "Formal Reduction of the General Combinatorial Decision Problem." Amer. J. of Math. 65: 197-215.

26. ______________. 1944. "Recursively Enumerable Sets of Positive Integers and their Decision Problems." Bulletin of the American Mathematical Society 50: 284-316.

27. ______________. 1947. "Recursive Unsolvability of a Problem of Thue." JSL 12: 1-11.

28. Rasiowa, H. and Sikorski, R. 1970. The Mathematics of Metamathematics. Third Edition, Monografie Matematyczne, Warszawa.

29. Roehrkasse, R. C. 1971. "Abstract Digital Computers and Automata." Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, Georgia.

30. Rogers, Hartley. 1967. Theory of Recursive Functions and Effective Computability. New York: McGraw Hill.

31. Sacks, Gerald E. 1960. "An Uncountable Set of Incomparable Degrees." Proceedings of the American Mathematical Society 11: 61-62.

32. ______________. 1963. Degrees of Unsolvability. Annals of Mathematics Studies, no. 55, Princeton, N.J.

33. ______________. 1964. "The Recursively Enumerable Degrees are Dense." Ann. of Math. 80: 300.

34. ______________. 1967. "Metarecursion Theory" in Sets, Models and Recursion Theory, edited by J. N. Crossley. Amsterdam: North Holland.

35. Schoenfield, Joseph R. 1958. "Degrees of Formal Systems." JSL 23: 382-392.

36. Scott, D. 1965. "Logic with Denumerably Long Formulas" in The Theory of Models, edited by J. W. Addison, L. Henkin, and A. Tarski. Amsterdam: North Holland.

37. Sikorski, Roman. 1969. Boolean Algebras. New York: Springer-Verlag.

38. Smullyan, Raymond. 1961. Theory of Formal Systems. Annals of Mathematics Studies, no. 47, Princeton, N. J.

39. Tarski, Alfred. 1948. "A Decision Method for Elementary Algebra and Geometry." Rand Corporation, Santa Monica, California.

40. Tarski, A., Mostowski, A. and Robinson, R. 1953. Undecidable Theories. Amsterdam: North Holland.

41. Tarski, A. and Scott, D. 1958. "The Setential Calculus with Infinitely Long Expressions." Colloquium Mathematica 6: 165.

42. Turing, Alan M. 1936. "On Computable Numbers with an Application to the Entschiedungsproblem." Proceedings of the London Mathematical Society, ser. 2 42: 230-265.

VITA

Joseph R. Horgan was born in Elizabeth, New Jersey, on February 8, 1942. Mr. Horgan received the Bachelor of Arts degree in Philosophy in 1964 and the Master of Arts degree in Philosophy of Science in 1970, both degrees from the University of Delaware. In 1972, Mr. Horgan received the degree of Doctor of Philosophy in Information and Computer Science from the Georgia Institute of Technology.

Mr. Horgan has taught logic and philosophy at the University of Delaware, and has been a research and teaching assistant at Georgia Institute of Technology. He has also been employed as a systems analyst by the International Business Machines Corporation and by the University of Delaware.